Izdaje BIGZ OOUR „Duga“

računari

u vašoj kući

Cena 200 dinara

5

April/maj 1985.

najbolji YU programeri pišu za vas

novi „komodor“, novi „atari“

Specijalno izdanje časopisa „Galaksija“

šesnaestogodišnjaci za računarom mladi genijalci ili izgubljena generacija

„galaksija“ fina grafika

listinzi
„spektrum“
„komodor“
„galaksija“
„amstrad“

„spektrum“ na svoju sliku i priliku

„komodor“ recepti za sprajtove

periferijska oprema
grafičke table

obrada teksta
pisanje bez muke

hakeri u nevolji

5 računari u vašoj kući

cena 200 D

april/maj 1985.

Specijalno izdanje časopisa „Galaksija"

Izdaje
Beogradski izdavačko-grafički zavod
OOUR Novinska delatnost „Duga"
11000 Beograd
Bulevar vojvode Mišića 17
Telefoni
650-161 (redakcija)
650-528 (prodaja)
651-793 (propaganda)
Generalni direktor
Gojko Zečar
Direktor OOUR „Duga"
v.d. Aleksandar Badanjak
Glavni i odgovorni urednik
Gavrilo Vučković
Urednik izdanja
Jova Regasek
Likovna i grafička oprema
Dušan Mijatović
Redakcija časopisa „Galaksija"
Tanasije Gavranović, pomoćnik glavnog i odgovornog urednika
Esad Jakupović, zamenik glavnog i odgovornog urednika
Aleksandar Milinković, urednik
Jova Regasek, urednik
Zorka Simović, sekretar redakcije
Srđan Stojančev, novinar
Gavrilo Vučković, glavni i odgovorni urednik
Stručna saradnja
Dejan Ristanović
Nevenka Spalević
Anđelko Zgorelec
Mihajlo Tešević
Autori tekstova
Predrag Bogdanović
Branko Đaković
Igor Fisher
Branko Hebrang
Vladimir Ilijevski
Milan Ivanović
Đorđe Janković
Vladimir Kostić
Srboljub Kuzmanović, dipl. ing.
Vojislav Lalić-Petrić
Ivan Nador, dipl. ing.
Damir Omrčen
Borivoj Perzić
Bogdan Petrović
Dejan Ristanović
Jelena Rupnik
Duško Savić
Jovan Skuljan
Mr Veljko Spasić
Milan Tadić
Dušan Veljković
Zvonimir Vistrička, dipl. ing.
Anđelko Zgorelec
Crteži
Miša Marković
Tehnička saradnja
Ljubiša Milovanović
Ljev Rjadčenko
Prevodioci
Esad Jakupović
Ksenija Pješčić-Lebedinski
Izdavački savet „Galaksije"
Dr Rudi Debijadi, prof. dr Branislav Dimitrijević (predsednik), Radovan Drašković, Tanasije Gavranović, Živorad Glišić, Esad Jakupović, Velizar Maslać, Nikola Pajić, Željka Perunović, prof. dr Momčilo Ristić, Vlada Ristić, dr inž. Milorad Teofilović, Vidojko Veličković, Velimir Vesović, Miloje Vuković
Štampa
Beogradsko izdavačko-grafički zavod
11000 Beograd, Bulevar vojvode Mišića 17
Žiro-račun kod SDK 60802-833-2463
Devizni račun kod Beobanke
60811-620-6-82701-999-01066
Za inostranstvo cena dvostruka (400 D, 2,50 US $, 6.50 DM, 45 Sch, 5,50 Sfrs, 20 Ffrs)
Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-77/72-03 i „Službenog glasnika" broj 26/72, ovo izdanje oslobođeno je poreza na promet.

sadržaj

3/ šta ima novo u svetu računara

8/ najbolji YU programeri
virtuozi na kućnom računaru

11/ računari u izlogu
dvoboj divova

13/ računari u razgovoru
mladi genijalci ili izgubljena generacija

16/ periferijska oprema
radost crtanja

20/ periferijska oprema
neki novi „epsoni"

22/ računari i programiranje
miladinova čarobna lampa

25/ računari i obrada teksta
pisanje bez muke

29/ računari iz mog ugla
pirati trče počasni krug

30/ biblioteka programa

37/ računari u akciji
strogo kontrolisani diskovi

41/ škola sistemskog programiranja
raspodela memorije

45/ umetnost programiranja
interapti u bejziku

48/ majstorije na računaru
na svoju sliku i priliku

50/ hajde da se igramo

53/ računari i ljudski um
zablude oko „elize"

56/ programiranje u bejziku
sprajtovi na „komodoru"

59/ u svetu slučajnih brojeva
zec iz šešira

63/ hakeri u nevolji
SOS

64/ računari u domaćoj radinosti
priča o finoj grafici

šta ima novo

Računari u domaćoj radinosti

Postojeći propisi u SR Hrvatskoj u oblasti zanatstva i kućne radinosti nakon nekih izmjena i dopunskih tumačenja sada dozvoljavaju bavljenje računarima i u kućnoj djelatnosti kao načinom privređivanja. Republički komitet za energetiku, industriju, rudarstvo i zanatstvo dao je dopunska tumačenja postojećeg Zakona o zanatstvu; u domaćoj radinosti — kao obliku privređivanja — mogu se izrađivati štampane pločice koje se koriste u elektronici, a u okviru zanatske djelatnosti pod nazivom „izrada predmeta od plastičnih masa i umjetnih smola".

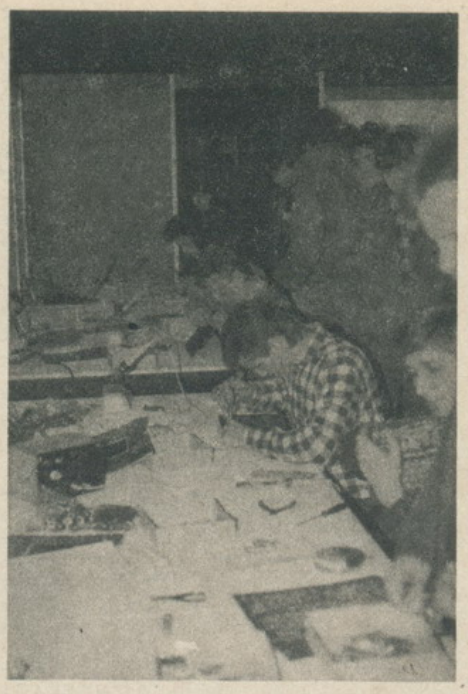
Ovaj Komitet je zauzeo i stajalište da se mijenja naziv djelatnosti „umnožavanje podataka za kompjutorske kartice i magnetne vrpce za elektroničku obradu podataka" tako da u buduće glasi: „elektronička obrada podataka". U okviru ove djelatnosti mogu se izrađivati programi za elektronička računala i mediji za pohranu podataka. Izrada programa i dorada m. dija za elektroničku pohranu podataka, po svojoj funkciji i načinu izrade, mogu se smatrati proizvodnjom, pa se ovi poslovi mogu obavljati i u domaćoj radinosti, jer za ovu djelatnost nije potrebno dokazivati stručnu spremu. Međutim, u okviru ove djelatnosti **ne mogu** se u SR Hrvatskoj pružati usluge obrade podataka, budući da se to ne smatra proizvodnjom.

M. Ivanović

Omladinski radio 101

Zagrebački Omladinski radio 101 je gotovo od samog početka svog emitovanja (maj 1984.) počeo posvećivati pažnju kompjutorima. Emisija posvećena kompjutorima, mikroelektronici i informatici naziva se „Skok u sadašnjost". U sklopu emisije emitiraju se i programi za kućne kompjutore, najčešće za Sinclair ZX Spectrum i Commodore 64. „Skok u sadašnjost" uređuju i vode ing Davor Žunić i dipl. ing Zvonimir Vistrička, uz pomoć stalnog suradnika Pere Petrovića.

Glavna zadaća emisije je popularizacija svih oblika korištenja kompjutora, informiranje slušaoca o kretanjima na tržištu, preporuke za korištenje literature, hardwarea i softwarea, stručni savjeti, itd. U studio se često dovode i gosti koji odgovaraju na pitanja vezana uz tu tematiku. Za vrijeme emitiranja atmosfera u studiju je opuštena, što stvara prisniji i neformalni odnos prema slušaocima koji se u okviru kontakt programa javljaju s pitanjima i prijedlozima.



Djelatnost emisije „Skok u sadašnjost" se nije zaustavila samo na radio-emisijama, već se proširila i na manifestacije izložbenog karaktera. U Zagrebu će se dugo pamtiti „Kompjutorska veselica" u organizaciji Omladinskog radija. To je bila multimedijska izložba kompjutorske grafike popraćena predavanjima, demonstracijom kompjutora, izložbom literature, te prikazivanjem svjetskih dostignuća na polju kompjutorske grafike — preko videa, diaprojekcija te izloženih grafika. Cilj izložbe je bio pokazati da kompjutori nisu samo sredstvo za matematičke proračune već i za ispoljavanje kreativnih osobina programera. Izložba je bila održana u divnom prostoru Društvenog centra boraca, RVI i omladine Zagreba početkom mjeseca novembra, 1984. godine.

Popularizirati kompjutore nikada nije dosta, a načini popularizacije moraju biti što raznovrsniji. Stoga autori emisije „Skok u sadašnjost" daju i razrađuju ideju o opće jugoslavenskom natječaju za najbolji originalni program, pod pokroviteljstvom Omladinskog radija 101. Nagradni fond je velik i sigurno će biti dobar radni poticaj za brojne domaće programere. Za one programere koji nemaju vlastiti stroj organiziran je i „Computer Workshop" Omladinskog radija 101 — radionica koja radi svakodnevno od 8 do 22 sata u predvorje jednog od najposećenijih kina u Zagrebu, kina Studentskog Centra.

Radionica (Workshop) je otvorena sredinom januara i ostaje otvorena do 1. juna 1985. godine, kada i završava natječaj. Oni koji nemaju vlastiti kompjutor mogu doći raditi na Orlovima, Spectrumima, Commodoreima, Teri, Partneru ili Appleu, a po želji i pohađati tečaj za brzo savladavanje basica ili pascala. Utorkom se održavaju predavanja s temom o kompjutorskoj grafici, a ujedno se publici demonstriraju kod nas rjeđi kompjutori kao (Amstrad/Schneider CPC 464, Sinclair QL, BBC). Za one koji sa svojim kompjutorima imaju nekih problema, subotom su postavljena dežurstva za otklanjanje kvarova.

Za poklonike „Galaksije" nedavno je održano zanimljivo natjecanje u brzom sastavljanju kompjutora „galaksija". Predstavnik firme „Mipro" iz Buja donio je 5 „galaksija" u kit obliku. Od velikog broja prijavljenih za natjecanje žrijebom je izabrano petoro kandidata koji su željeli dokazati svoje znanje iz praktične elektronike te osvojiti vrijednu nagradu — najbrži svoju sklopljenu „galaksiju". Uz brzinu, uvjet je bio da „galaksija" proradi.

Prvi od petorice je bio gotov Tomislav Jurlin, student Elektrotehničkog fakulteta, ali, na žalost, njegov kompjutor nije proradio. Pobjednik, osvajač nagrade Lenz Darko je nakon višesatnog lemljenja došao do vrijedne nagrade, najprihvatljivijeg domaćeg računala. Lenz je, također, student ETF-a i nema sumnje da će mu kompjutor dobro doći.

Ovakav način predstavljanja kompjutora potakao je velik interes za „galaksiju", koja je time u Zagrebu stekla puno više poklonika i simpatizera. Nadajmo se da će ovakvih atraktivnih prezentacija biti još više, naravno iz kvalitetan predgovor o kompjutoru, kakav je, naprimjer, pružio predstavnik „Mipra"

Zvonimir Vistrička

Megabit na jednom čipu

Za vreme najnovije konferencije, posvećene fizici čvrstih tela, koja je održana u San Francisku, japanske firme Tošiba, Hitači, NEC i NTT podnele su izveštaj o proizvodnji novog silicijumskog čipa, koji može da memoriše milion jedinica informacije — četiri puta više nego što je dosad bilo moguće. Čip je debljine 40 mikrometara (što znači da je tanji od jedne vlasi kose), površine koja odgovara veličini pera za crtanje i sadrži 800 metara provodnog materijala. Na njemu se može memorisati 128000 alfanumeričkih znakova, a to je ravno obimu knjige od oko 75 stranica. Američki proizvođači najavljuju proizvodnju ovakvog čipa za početak ove godine. Smatra se da će za nekoliko godina uspeti i razvoj čipa od 4 megabita.

Što su dimenzije čipa manje, to materijal od kojeg je napravljen mora biti čistiji. Na konferenciji je predstavljen i uređaj za ispitivanje čistoće silicijumskih slojeva koji za manje od jedne sekunde može da ustanovi prisustvo nepoželjnih čestica 100 000 puta manjih od debljine vlasi kose.

Stolica za kompjuteraše

Hakeri koji zbog dugotrajnog sedenja dobijaju bolove u leđima mogli bi najzad da odahnu — barem u Engleskoj. Na tržištu, naime, nedavno se pojavila jedna stolica koja se uspešno nosi sa bolom u leđima. Problem, inače, nije ni malo naivan. Zbog tog sindroma samo u Velikoj Britaniji gubi se 29 miliona radnih dana godišnje!

Nova obrtna stolica ima dvostruki naslon, čiji nagib se prilagođava za različite vrste rada, ili se fiksira u raznovrsnim položajima. Visina sedišta reguliše se pneumatski, uz pomoć dugmeta, čime se izbegava nepovoljan uticaj na kičmu kod niskog sedenja. Meki, iskošeni prednji rub sedišta omogućuje bolje prenošenje težine tela na noge bez ometanja cirkulacije.

Stolica se, kao što vidi, ubila za kompjuteraše. Na žalost, košta koliko i manji kućni računar — a kod nas ne može ni da se uveze. Hakerima, zasad, ostaje da češće prekidaju rad sa računarom i više se kreću.

Epl za sva vremena

U 1984. godini Apple*° je predstavio dva nova kompjutera — „mekintoš" (Macintosh) i IIc. I dok se šire glasine da trećerazredne kompanije proizvode kolor verziju „meka" i da se azijska šrafciger industrija sprema da izađe na američko tržište sa kopijom „mekintoša" po ceni od 1000 dolara, „Apple" vadi novog keca iz rukava pod imenom „epl IIx". Prinova u familiji ima visok stepen kompatibilnosti sa dosadašnjim modelima iz serije II, ali istovremeno, predstavlja i značajan korak napred u razvoju nove tehnologije. Rešenje zagonetke leži u malom komadu silicijuma koji na sebi nosi broj 65816.

Mikroprocesor 65816 je 16-bitna, CMOS (komplementarna metal-oksid tehnologija) verzija klasičnog 6502 mikroprocesora koji je mozak u „eplu II". U emulacionom modu 65816 je rasporedom izvoda i softverski potpuno kompatibilan sa starim 6502--uticajem na samo jedan njegov fleg, 65816 postaje po našoj želji, osmobitni 6502. Pod punom ratnom spremom, međutim, on raspolaže moćnijim setom instrukcija i može direktno da adresira do 16 megabojta radne memorije. I pored kritika da će sa ovako velikim memorijama čist i delotvoran program odlazi u muzej voštanih figura, ne možemo zaobići činjenicu da ovako veliki memorijski kapaciteti otvaraju nove mogućnosti za programere sa velikim idejama.

Zahvaljujući tome što 65816 može da oponaša 6502, programima napisanim za predhone „eplove" kompjutere nisu potrebne nikakve izmene da bi radili na IIx-u. Ne zadovoljavajući se samo sa kompatibilnošću, inženjeri Western Design Centers-a su povećali set instrukcija u odnosu na 6502, dodali nove adresne modove i proširili mogućnost rada sa prekidima. Zbog ovakvih osobina, ovaj procesor se može upotrebiti kod već postojećih sistema uz male ili nikakve izmene.

I dok je malo verovatno da će Apple nuditi proširenja za stare kompjutere iz serija II, IIx će sa 65816 i 256 kilobitnim memorijskim čipovima koriste se i u proširenom „mekintošu" sa 512 K, izaći na tržište sa 512 K u standardnoj verziji i mogućnostima velikih memorijskih proširenja.

Jedina stvar koja kvari ovu lepu sliku i dovodi u pitanje kompatibilnost novog „epla" sa dosadašnjim dvojkama je disk drajv. Dosadašnji 5,25 inčni diskovi korišćeni na „Apple"-ovim računarima postaju tehnološki zastareli. Očigledno je da će Sonijevi 3,5 inčni diskovi upotrebljeni u „meku" postati industrijski standard i zameniti 5.25 inčne diskove, kao što su ovi zamenili 8 inčne dinosauruse iz 70-tih godina. Ako IIx želi da drži korak sa vremenom, onda bi morao da bude upotrebljen disk drajv od 3,5 inča, ali čemu onda softverska kompatibilnost kada sa postojeće diskete ne možete učitati program u novi računar? Rešenje je u tome što će IIx-u biti vraćeni slotovi sa „epla IIe" koje IIc nije imao i tako omogućiti korišćenje dodatne disk jedinice od 5,25 inča, pa samim tim i upotrebu i/ili kopiranje programa na postojeći drajv od 3,5 inča.

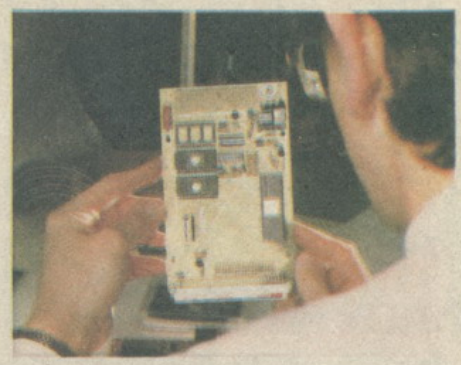
Malo je verovatno da će se IIx javiti u prodaji u prvom kvartalu ove godine zbog toga što bi hiljade onih koji su u toku božićnih praznika kupili IIc i IIe bili dovedeni u nezavidan položaj, jer bi njihov tek kupljen računar u neku ruku već bio zastareo. Praktično, datum kada će IIx biti izbačen na tržište zavisi od prodaje ostalih modela iz serije II.

V. Lalić-Petrić

Sintetizator za sve jezike

Novi sintetizator govora sa oznakom TDS—910 može elektronskim putem da podražava bilo koji jezik i dijalekt, a priključuje se na čitav niz uređaja, uključujući i sisteme obezbeđenja i kontrolne uređaje u industriji, kako bi davao poruke „skrojene" po zahtevu korisnika.

Projektant Pitr Raš (Peter Rush) (na



slici ispituje ploču štampanog kola) prilagodio je sistem standardnom „Eurocard" formatu, kako bi dobio široku primenu u elektronskoj industriji.

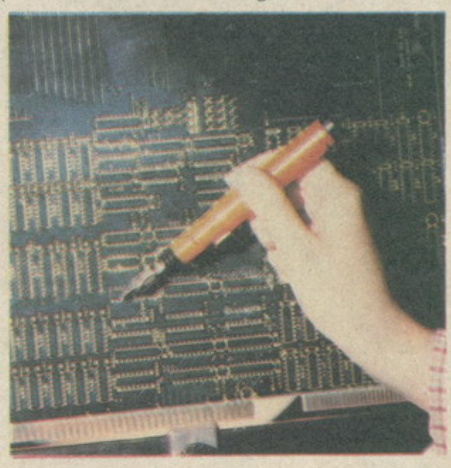
Sistem se može koristiti za davanje specifičnih informacija u situacijama u kojima je neophodna složena kontrola, na primer u fabrikama i na morskim platformama za naftu. Poruke upozorenja mogu se, u urgentnim situacijama, automatski prenositi telefonom do kuće rukovodioca, pružajući tačne podatke u stilu „vatra u mašinarnici", „poplava u suterenu" ili „otvoren sef".

Govorni sintetizator isporučuje se potpuno programiran prema rečniku korisnika. Postupak je jednostavan, jer se koriste reči izgovorene u mikrofon priključen na jedan specijalno projektovani kompjuter koji dozvoljava sve potrebne izmene u tipu glasa pre nego što se on definitivno prenese do memorije sintetizatora.

Pojedinačni delovi — govorni sintetizator, govorna memorija i integrisana kola — mogu da se dobiju jeftino, a kupac ih sam može sastaviti. Proizvođač je firma „Triangle Digital Servise" (23 Campus Road, Walthamstow, London).

Alatka iz sveta snova

Zove se malo neobično — „Orix Portasol" — a služi „za svako doba i svako mesto". Reč je o jednoj od najmanjih lemilica, za trenutne popravke štampanih veza. Ova zanimljiva prenosiva alatka radi na pricnipu novog i jednostavnog hemijskog procesa. Kao izvor zagrevanja koristi se tečni gas koji se nalazi u rezervoaru u unutrašnjosti lemila. Prema tvrdnji proizvođača, firme „Greenwood Electronics" (Portman Road, Reading, Berkshire, En-



gland), lemilica može da obavi raznovrsne poslove za kratko vreme — bez obzira da li lemilicu koristi profesionalac ili amater.

Nešto manji od flomastera, bez kablova ili električnog vrha, „Portasol" ima platinski katalizator u vrhu za lemljenje i može da radi čitav sat sa unutrašnjom rezervom butana. Patentirani katalitički konverter obezbeđuje zagrevanje vrha, tako da nema plamena u toku lemljenja. Brzina konverzije može da se podesi tako da temperatura vrha iznosi između 250°C i 450°C.

Pošto „Portasol" pruža energiju kao električna lemilica od 60 W, može da se koristi i za kompletne poslove lemljenja. Lemilica, čije dimenzije iznose samo 175 mm × 19 mm, može da se napuni u roku od nekoliko sekundi, a postupak je isti kao kod punjenja gasnog upaljača za cigarete. Lemilica se isporučuje zajedno sa zaštitnom kapicom u koju je ugrađen kremen. Odgovara svim sigurnosnim standardima, a izvor gasa kao izvor energije onemogućuje električna oštećenja osetljivih komponenti. Svi zamenljivi delovi, uključujući i konvertor, mogu se dobiti na tržištu.

„Epson" u Ljubljani

„Mladinska knjiga" iz Ljubljane počela je krajem februara s prodajom prenosivih računara „epson" s oznakom HIS—5 s ugrađenim mini-printerom, mini-ekranom i mikrokasetofonom — po ceni 380.000 din. U prodaji su i dva tipa printera: R-80 F/T+ (format A4 — 280.000 din) i R-100 (format A3 — 380.000 din) s dodatnom opremom (interfejsi, kablovi, priključci, programi) po izboru. Uz računar i printer moguće je kupiti i odgovarajuću dovstruku disk jedinicu TF-20 (655 K — 380.000 din).

Računarska revolucija u Kini

Računarska revolucija u Narodnoj Republici Kini je zaista u punom zamahu.

Sedmogodišnjim planom razvoja mikroračunara u NR Kini planira se proizvodnja koja za 1985. treba da obezbedi 500 mikroračunara, da bi se 1990. broj mikraća popeo na 10.000, a već dve godine kasnije i na čitavih 40.000 komada. Proizvodnju mikroračunara treba istovremeno da prati i proizvodnja odgovarajućih čipova (uglavnom RAM-ovi do 16 K, ROM-ovi od 4 i 8 K i neki od osmobitnih procesora). Visokoserijska proizvodnja čipova je osvojena.

Dva poznata američka proizvođača računara, angažovana su na izgradnji dve fabrike koje, pored povećanja ukupnog obima proizvodnje računara, treba i da omoguće i prodor novih, naprednijih tehnologija.

Nekoliko singapurskih i hongkonških firmi koristi pogodnosti jeftine kineske radne snage i investira u kineske pogone. Ovaj program podržava i kineska vlada jer na taj način strane tehnologije ulaze „na mala vrata".

Zbog geografskih pozicija Singapura, Hongkonga — a ne treba zaboraviti ni Maleziju, Filipine, Indoneziju, zemlje gde se čipovi danas najviše proizvode — odlučeno je da se „Silicijumska dolina" NR Kine sagradi u južnim provincijama.

Svetska banka je odobrila namenski kredit za kupovinu računara u vrednosti od 38 miliona dolara. To je, verovatno, jedan od razloga što je „Hewlett — Packard" otvorio svoje predstavništvo u Pekingu. A Pakardov primer, nema sumnje, slediće i drugi veliki proizvođači, tim pre što je kinesko tržište veliko, a NR Kina je poznata kao solidan platiša.

Nepravedno bi bilo pominjati samo planove, a ne i ono što je već postignuto. Trenutno se u desetak fabrika već proizvode veliki, mini i i mikro računari. Skoro stotinak kooperanata proizvodi delove periferijske opreme i podsklopove. Proizvode se računari koji liče ili su kompatibilni sa DEC PDP 11/34 i IBM 370. Priča se, takođe, da kineske inženjere i tehničare od proizvodnje IBM 4300 deli oko pet godina.

Nisu zaboravljeni ni kadrovi — više hiljada studenata studira na zapadnim univerzitetima. Istovremeno, kineska vlada je uputila poziv svim poznatim stručnjacima iz ove oblasti, kineskog porekla, da se vrate u otadžbinu i pomognu bržem razvoju.

U okviru značajne trgovinske razmene sa Sjedinjenim Američkim Državama, koja je u velikoj ekspanziji, Kina u odnosu na ostale zemlje tzv. „realnog socijalizma" ima poseban tretman u oblasti računara i uvoza vrhunske tehnologije. Dok je za ostale izvoz vrhunske tehnologije iz SAD uglavnom pod embargom, za Kinu se primenjuje drugi „aršin". Prema izjavi ministarstva trgovine SAD, po odobrenju predsednika Regana dozvoljava se izvoz naprednih tehnologija iz oblasti računara i elektronike. Ipak za svaki pojedinačni izvoz određene tehnologije potrebna je posebna dozvola. Na taj način je ostavljena mogućnost da do izvoza nekih vrhunskih tehnologija ipak ne dođe, barem ne dok malo ne — zastare.

J. Nador

Optički diskovi

Takozvani „Desk top personal computer" (lični računari koji staju na pisaći sto) koriste se već duže vremena u modernim kancelarijama zapadnih zemalja. Ipak, gomile papira i registratora sa podacima još uvek se ne smanjuju. Flopi i hard diskovi mogu da memorišu 400 K i 10 MB respektivno. To zvuči mnogo, ali ako se sav ovaj prostor iskoristi za memorisanje teksta, onda na jednu disketu stoje samo 150 strana teksta (predpostavka je da u redu ima 80 karaktera a da na stranu staju 33 reda i da se piše samo sa jedne strane). U slučaju da se koristi hard disk, situacija je znatno bolja — može se smestiti čitavih 2780 strana teksta, ali priznaćete da za jednog referenta ni ta cifra nije impozantna.

Ako insistiramo na ličnom računaru koji je nezavistan u odnosu na neki veliki sistem, onda u ovom trenutku to može da mu obezbedi samo laserski disk, koji može da memoriše od 1 do 5 gigabajta! Prevedeno na strane teksta, to iznosi 278 600 kucanih strana. Da stvar bude lepša zajedno sa tekstom, moguće je povezati i sukcesivno memorisati i odgovarajuće digitalizovane slike (fotografije i sl.).

Ključni deo optičkog disk drajva je precizno kontrolisani laser male snage koji može da upisuje i iščitava podatke sa specijalnog, plastikom presvučenog aluminijumskog diska. Ova sprega elektronike i laserske tehnologije omogućava smeštaj veoma velike količine informacija na ekstremno malom prostoru.

Kako se upisuju podaci? Posto nam je poznato da podatak može biti ili nula ili jedinica, lako se dolazi do zaključka da se jedan od ta dva podatka upisuje na disk termičkim dejstvom lasera na površinu aluminjumskog diska, dok se drugi podatak ne upisuje. Upisani podatak se može videti kao tačkica — udubljenje u disku mikronskih dimenzija.

Na žalost, jednom upisani podaci se ne mogu brisati. Prema tome, u ovom momentu laserski disk je, pre svega, namenjen za masovno memorisanje informacija koje su trajnog karaktera — dakle memorisanje arhive.

Kako stvari stoje, prvi računar koji će se moći da pohvali da ima mogućnost memorisanja podataka na laserski disk je IBM PC. Za sada još uvek visoka cena (17000 do 19000 dolara), verovatno će u bliskoj budućnosti biti oborena međusobnom konkurencijom više potencijalnih proizvođača.

Industrijski standard za laserski disk, razume se, još uvek ne postoji. Pominju se dimenzije od 3 do 14 inča u prečniku.

Ipak, sve ovo ne znači da Flopi i hard diskovi odlaze u staro gvožđe. Njihova namena će biti samo malo preciznije određena.

I. Nador

BBC C ili model C

Iz pomalo zagonetnih nastupa direktora Krisa Karija (Criss Curry) i Kristofera Varda (Christopher Ward), novinari su zaključili da Ejkorn sprema računar koji se zove Model C. Ovo, po svemu sudeći, nije BBC C već „Comunicator", računar koji će konkurisati ICL-ovom OPD-u (kompjuteru koji je baziran na Sinklerovom QL-u).

Model C će biti zasnovan na 65C816, šestaestobitnom procesoru koji je skoro u pin kompatibilan sa 6502. Model C će u startu imati 256 ili 512 K RAM-a, koji će moći da se proširi bar do jednog megabajta. Kompatibilnost sa BBC serijom će biti stoprocentna, pošto 65C816 može da radi u modalitetu u kome emulira 6502 (tada, jasno, ne može da adresira više od 64 Kb memorije).

Cena novog modela je nepoznata, ali je Kari pominje sumu od 500 do 800 funti, u koju je uračunata jedna disk jedinica. Datum izlaska na tržište je, naravno, velika tajna, ali nam se čini da model C ne treba očekivati previše brzo.

D. Ristanović

Watfordovi noviteti

Jugoslovenskim vlasnicima BBC-ja i Electrona dobro poznata prodavnica Watford Electronics (250 High Street, Watford, England) neprekidno proširuje asortiman svojih proizvoda. Posle veoma uspelog „Single Density" disk interfejsa, pojavila se i „Double density" varijanta koja je zasnovana na Intelovom kontroleru 8272. Uz sve opcije koje je posedovao dosadašnji interfejs (disk sektor editor, 64 umesto 32 programa na svakoj strani diskete, prebacivanje softvera sa kasete na disk, programi za formatiranje i verifikaciju u ROM-u, obilato korišćenje džoker znakova i mnogo što-šta drugo), dodate su naredbe za prepisivanje diskova u raznim formatima upisa i ugrađen veoma kompletan 8271 emulator koji omogućava izvršavanje svih zaštićenih programa koji se prodaju na disku, čak i „Elite". Cena je 85 funti.

Watford Electronics prodaje i tablu za proširenje RAM-a od 32 K. Ovaj RAM može da se koristi kao video memorija, bafer za printer ili bilo koji drugi bafer. Uz umerenu cenu od 65 funti (BBC sa ovom tablom košta samo 360 funti+VAT), ova ekspanziona tabla predstavlja zanimljivu alternativu dodatnom procesoru. Mana ovoga proširenja je što se dodatna memorija ne može iskoristiti za smeštanje ROM modula.

Za one koji ne žele da nabave Solidisk, Watford Electronics od ovog meseca prodaje 16 K disk RAM. Priključujemo ga umesto jednog od ROM-ova i „punimo" softverom sa diska. Tako ne moramo da kupujemo skupe EPROM-e da bismo pirasovali „View", „Wordwise" ili „Pascal", dovoljno je da jednom odvojimo 40 funti za ovaj dodatak. Sadržaj RAM-a se čuva čak i dok je disk isključen zahvaljujući ugrađenom pakovanju baterija.

Vlasnici BBC-ja koji imaju dosta ROM-ova a ne žele da troše novac na ROM tablu mogu da kupe ZIF priključak. Radi se o Tekstolovom podnožju koje se montira sa desne strane tastature računara i koje povezujemo sa jednim od ROM podnožja u računaru. U ovo podnožje možemo da smeštamo bilo koji EPROM bez rizika da izazovemo njegovo oštećenje nestručnim rukovanjem. Cena je prava sitnica — 18 funti.

Dejan Ristanović

Računari sasvim lično

Dinar vs commodore

U petak 16. marta 1985. „Komodor 64" se umorio od naporne utrke sa jugoslovenskom nacionalnom valutom — sveži i za trku raspoloženi dinar je pretekao ovaj umorni računar. Počev od 16. marta, naime, beogradska carina zaplenjuje „komodor 64" koji sada košta više od 40.000 dinara. Nastavak ove vesti očekujte u sledećim „Računarima" — dinar možda ima dovoljno snage da pretekne i za sada nedostižnog „spektruma"!

Novo u kioscima. . .

Iako je računara koji se uklapaju u dozvoljeni uvozni limit sve manje, domaći hakeri nemaju razloga da se požale na broj kompjuterskih časopisa koje mogu da pronađu u kioscima. Od početka ove godine pojavili su se časopisi MR (u izdanju Sportske tribine, Zagreb, treba da izlazi jednom mesečno), TREND (specijalno izdanje SAM-a sa istom frekvencijom izlaženja), i Pilot Video (specijalno Kvizovo izdanje posvećeno kompjuterskim igrama). Haker koji želi da prati sve domaće časopise treba svakoga meseca da izdvoji otprilike 1.300 dinara, što čini oko 15.000 dinara godišnje. Saznajemo da će redakcije svih naših časopisa pokušati da zaštite džepove svojih čitalaca tako što će, sledeći svetle primere „Računara u vašoj kući", redovno kasniti!

. . . I novo u oglasima

Dok se broj časopisa lagano umnožava, broj kompjuterskih klubova to radi brzo — čitaoci oglasa su svakako primetili da kompjuterski „klubovi" niču kao pečurke posle kiše. Jeste li znali da su se pojavili Verify Ne$a, Sinclub, Bros Software, King Byte, Bugy Club, Spectrum House, New Data, White Lightning, Videosoft, Joker Software, Royal Service i, posebno interesantni, Giga's Software? Sa žaljenjem vas obaveštavamo da su surovi tržišni uslovi učinili da je čak i gola egzistencija klubova Sunnsoftware, Comet Software, Intersoft, Columbia Software, Computerland, Spectrum goes to Hollywod, Byte Shop i Tangram Software dovedena u pitanje. Čuvene domaće softverske firme Joystick Club i Orion Software i dalje drže primat na tržištu prvenstveno zahvaljujući visokoj produktivnosti i ulaganju u prošireni reprodukciju.

Čime se bave svi ovi klubovi? Najbolje je da vam, umesto odgovora, navedemo ime jednoga od onih koji su propali zbog iskrenosti: „Software Break&Co".

Napomena: ako su neki od gornjih naziva pogrešno spelovani, obratite se njihovim vlasnicima koji, valjda, najbolje znaju kako im se klub zove!

Dejan Ristanović

Razglednice iz Londona

Anđelko Zgorelec

Spas u poslednji čas

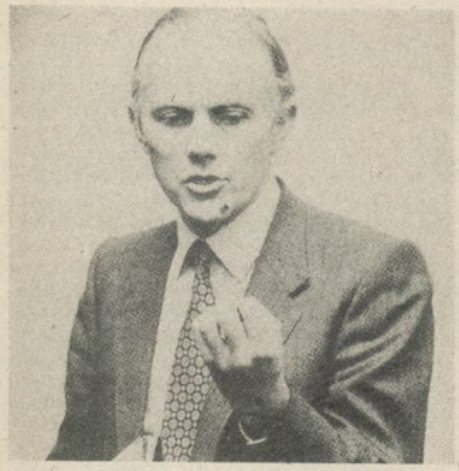
Među stotinu putnika koji su se u petak 15. februara poslednjim avionom vraćali iz Milana za London, dvojica muškaraca srednjih godina bili su, sigurno, posebno srećni. Posle višednevnih pregovora u tom najvećem poslovnom centru Italije, njihovi napori su konačno urodili plodom. Jedna od najpoznatijih britanskih mikroračunarskih kompanija koju su upravo oni stvorili i koja proizvodi jedan od najpopularnijih računara u ovoj zemlji, imaće osiguranu budućnost. Firma se zove Acorn, a zadovoljni putnici su njeni osnivači: Kris Kari (Chris Curry) i Herman Hauser (Hauser).

Naime, kao što smo javili u martovskoj „Galaksiji", postojala je velika opasnost da „Acorn" ode na bubanj — prošlogodišnji poslovni rezultati bili su više nego katastrofalni. Računari se sada toliko slabo prodaju da novac gotovo i ne ulazi u kasu ove kompanije. Zato je preduzeta brza akcija da se nađe neki finansijski bolje stojeći partner. Bilo je zainteresovano više velikih britanskih kompanija, ali predstavnici poznate italijanske firme „Olivetti" su bili najbrži. Za ulog od samo 10,4 miliona funti „Olivetti" je dobio 49,3% deonica u ovoj kompaniji i time stvarnu kontrolu nad proizvodnjom BBC računara. U Ivrei, blizu Milana, Glavnom štabu Olivettija sigurno zadovoljno trljaju rukama, jer Acorn je samo pre šest meseci vredeo gotovo 100 miliona funti! Povoljne vesti iz Italije popraćene su sa zadovoljstvom i u Kembridžu, Ejkornovom glavnom štabu. Doduše, Kris Kari i Herman Hauser gube kontrolu nad svojom kompanijom, ali budućnost Acorna osigurana je i, što je najvažnije, i dalje će se proizvoditi BBC-jev čuveni računar, ali sada uz podršku moćnog „Olivettija" i njegovog novca.

Interesantno je spomenuti da je „Olivetti" pre samo šest godina bio i sam gotovo na rubu propasti, kada je na njegovo čelo došao sposoban Karlo de Benedeti (Carlo De Benedetti) i preporodio kompaniju. Od proizvodnje odlično dizajniranih (izložene su i u Muzeju moderne umetnosti u Njujorku) ali zastarelih pisaćih mašina, „Olivetti" je prešao na proizvodnju modernih računara, koji se odlično prodaju. Benedeti ulaže novac i u manje elektronske i softverske kompanije, poznate po inovacijama, a usko se sarađuje i sa američkim telekomunikacijskim gigantom AT&T, koji drži 25% deonica u „Olivettiju". Jasno je da ove globalne veze mogu samo da pomognu Acornu. S druge strane, tvrdi se da je Benedeti bio naročito impresioniran ekipama mladih stručnjaka, koji rade za Acorn u njegovim centrima za razvoj u Kembridžu i Palu Altu u Silicijumskoj dolini u Kaliforniji. Od tih mladih ljudi traže se novi proizvodi, koje bi „Olivetti" preko svoje mreže plasirao po čitavom svetu. A Kris Kari i Herman Hauser neće više voditi kompaniju; oni odlaze u pozadinu i biće zaduženi za razvoj novih proizvoda.

Amstrad je kriv za sve

Kriza u svetu kućnih računara nije pogodila samo „Acorn". I drugi osećaju posledice slabe prodaje računara. Nedavno je objavljeno da Sinkler za mesec dana neće preuzimati računare iz fabrike koje proizvode za njega ne bi li smanjio zalihe u skladištima. I „Apple" je primoran na sličan korak: na kraće vreme zatvara svoje fabrike u Kaliforniji i Irskoj, a osoblje šalje na prinudni odmor. Iz Japana javljaju da su proizvođači MSX računara znatno smanjili obim proizvodnje.



Lud za špagetima: Aleks Reid, novi čovek na čelu Acorna

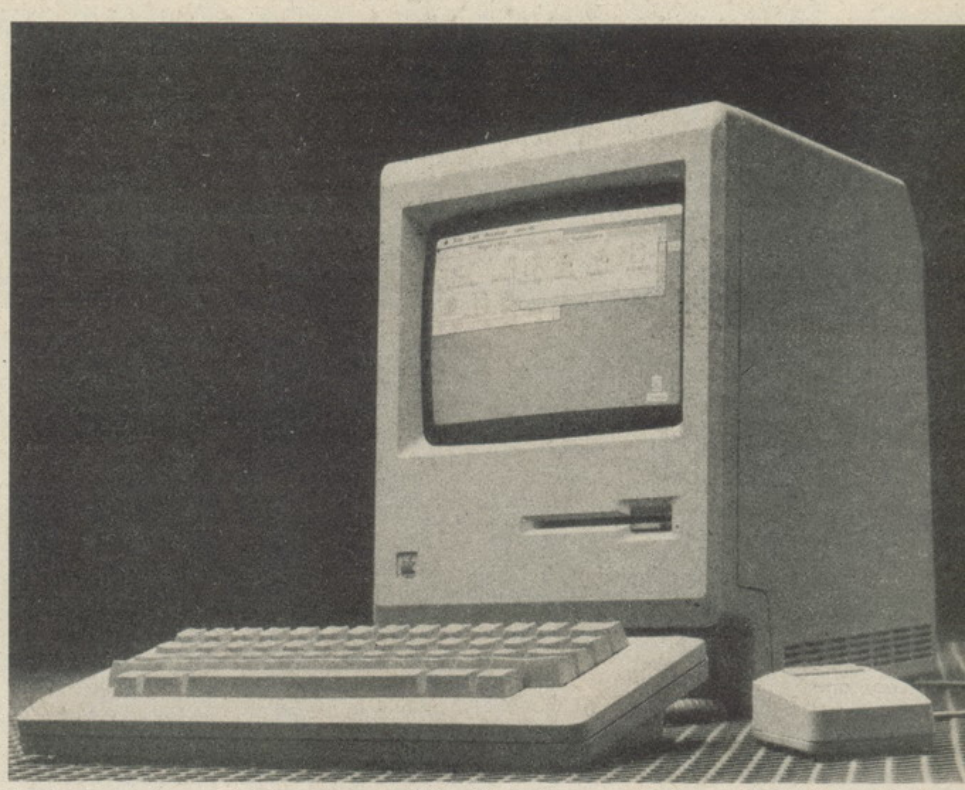
Jedine svetle tačke u ovom trenutku u Britaniji su za kućne računare firma „Amstrad", a za male poslovne sisteme firma ACI, koja proizvodi čitavu seriju računara i koji su ovde velika konkurencija moćnom IBM-u i njegovom PC-u. Mada „ACI" računari nisu kompatibilni sa PC-em, upotrebljavaju isti operativni sistem, pa je bilo relativno lako osigurati za njih mnoštvo softvera.

Na području kućnih računara „Amstrad" vodi po povoljnoj ceni, za specifikacije koje se nude. Za samo sedam meseci u Britaniji je prodato 200 hiljada modela CPC 464, što je navelo jednog komentatora da napiše da je uspeh „Amstrada" glavni „krivac" za debakl Acorna, jer su pojavom „Amstrada" na tržištu Acornovi računari počeli odmah slabije da se prodaju.

„Mek" i QL nakon godinu dana

Od svih novih proizvoda u prošloj godini, dva računara pobudila su izuzetnu pažnju, jer su, zbog izuzetnih specifikacija i 32-bitnih procesora — predstavljali prekretnicu na području personalnog računarstva. Applov „mekintoš" i Sinklerov Q1. No, izgleda da ovi proizvodi nisu ispunili one nade koje su ove dve firme polagale u njih.

Računari se, pre svega, relativno slabo prodaju — uglavnom zbog nedostatka dobrog softvera. I pored ogromnih svota novca koje je „Apple" trošio na oglašavanje, „Mek" u Americi ide osrednje, više od polovine mašina je prodato uz znatno sniženu cenu studentima, novinarima, prodavačima u mikroračunarskim radnjama. Najveći i najvažniji segment tržišta — poslovni



ljudi — nerado se odlučuje za „mek". Ni u Britaniji situacija nije bolja, jer se ovaj računar smatra preskupim. Jedino iz Francuske javljaju da „mek" ide dobro. Dopisnik časopisa „Infoworld" smatra da je glavni razlog to što Francuzi uvek staju na stranu slabijih, to jest „Appla" u borbi sa IBM-om. „Apple" je učinio zaista dosta na propagiranju ovog računara u Francuskoj, a postoji i obilje originalnih francuskih programa (neki od njih se, čak, prodaju i u Americi) za „mekintoš".

U Britaniji prvi rođendan Q1-a nije prošao nezapaženo. Kompanija Sinkler je organizirala izložbu, na kojoj su uzele učešće 24 firme sa 50 novih softverskih proizvoda i više novih perifernih jedinica. Organizator izložbe je, očigledno, želeo da pokaže da Q1 uživa podršku i većeg broja nezavisnih proizvođača. Od softvera je najviše bilo takozvanih korisničkih (utility) programa, novih jezika (pascal, forth i lisp) i dosta poslovnih programa za manje firme, koji bi trebali pomoći u knjigovodstvu, financijskom planiranju, i jednostavnijem vođenju knjiga za porezne organe.

Što se tiče hardvera, čak tri firme su pokazale disk jedinice za Q1, a najavljuje se i tvrdi disk. Zatim pokazano je više modema, te palica za igre, kao i laserski štampač.

Posle izložbe, „Sinkler" je organizovao i jednodnevni seminar za nezavisne proizvođače i novinare. Glavni govornik je bio ser Klajv Sinkler, koji smatra da, i pored sadašnjih poteškoća, Q1 ima sjajnu budućnost. Do sada je, po njegovim rečima, prodato 50 Q1-a i prodaće se do kraja ove godine još 200 hiljada. Izrazio je naročito zadovoljstvo što je vodeći škotski univerzitet Stratklajd (Strathclyde) odlučio da nabavi sedam hiljada Q1-a za sve svoje studente. Sinkler je, također, najavio i dva nova proizvoda — prenosni „spektrum" (!) koji će biti potpuno kompatibilan sa sadašnjim „spektrumom", kao i silicijumsku disk jedinicu — originalni Sinklerov proizvod u kome će čipovi preuzeti ulogu Flopi diska. Novi proizvod trebalo bi da se prodaje po ceni od 300 funti, a imaće kapacitet od pola megabajta.

Pet monitora za „Kung Fu"

Prvih meseci u godini obično se ne izdaju bolje igre. No ovih dana jedan program — igra pobuđuje dosta pažnje u kompjuterskim časopisima i dobija najbolje ocene. To je program „Kung Fu", koji je sastavio Beograđanin Duško Dimitrijević, a koji je nedavno poznata softverska kompanija „Bug Byte" pustila u prodaju u Britaniji. Svi prikazi u časopisima daju ovoj igri odlične ocene, a vodeći nedeljni kompjuterski list „Popular Computing Weekly" dao joj je svoje najveće priznanje — pet monitora. Stalni saradnik ovog časopisa Džon Minson piše ovako:

„Stavio sam ovaj program u memoriju kompjutera sa strepnjom — ne da će tipke na mom „spektrumu" ispasti zbog udarca, nego da bi to moglo biti najgluplja simulacija jednog sporta. Jer, šta je Kung Fu bez fizičkog napora, škripe kostiju i krikova „Aaaa".

Posle šezdeset minuta postao sam kao drogiran. Uz podršku neprekidne orijentalne melodije moj borac, veličine trećine ekrana i odlično animiran, kretao se nazad i napred, pokušavajući da obori svog kompjuterom kontrolisanog protivnika. Sa samo dva ručna udarca i dva pokreta nogu, on je blokirao, susdržavao, zatim našao priliku i udario. Protivnik je bio oboren i ja sam zaradio žuti pojas."

Džon Minson dalje piše da je i njegova prijateljica, koja je karate ekspert, oduševljena igrom. Samo četiri različita pokreta boraca je, smatra on, možda siromašan repertoar, ali igra je ipak dovoljno raznovrsna i zanimljiva, poput najboljih igara iz arkada. Džon zaključuje svoj napis ovako: „Ne nema sumnje da je to odličan program, i sve što mogu kazati je . . . Haj-i-a — (povik boraca Kung Fu-a)".

virtuozi na kućnom računaru

Najbolji YU programeri

Prvonagrađeni rad „Hiper BASIC" je izvanredan sistemski program koji će, ukoliko nas ne varaju iskustva stečena u toku postojanja „Galaksijinog" Kataloga programa, obradovati sve prave hakere. Poput u svetu čuvenog beta bejzika, „Hiper BASIC" proširuje „spektrumov" programski jezik, ali to nikako nije sve što ovaj program nudi. Igor Fischer je, očigledno, dobar poznavalac forta, jezika koji omogućava korisniku da sam proširuje sistemski softver svog računara, dodajući reči koje predstavljaju nove naredbe. „Hiper BASIC" za dodavanje novih naredbi koristi službenu reč DEF. Po startovanju programa možemo, na primer, da otkucamo !DEF "WAIT-CR", "PUT$" "":+CHR$ 13+"":": WAIT:PUT 1000: GSB i pritisnemo ENTER. Tako smo definisali novu naredbu WAIT-CR koja čeka da korisnik pritisne ENTER, a zatim javlja OK i emituje kratak ton. Ako zatim otkucamo !VLIST (još jedna reč pozajmljena iz forta), videćemo da je naša nova naredba potpuno ravnopravna sa svim naredbama „Hiper BASIC-a".

„Hiper BASIC" je, dakle, opremljen i gomilom unapred definisanih naredbi koje će neobično obradovati sve početnike koji se neće lako usuditi da dodaju reči koje su im potrebne. Neke od tih predefinisanih naredbi su: DEL, PUT, PUT$, ONERR, RST, DPEEK, DPOKE, GTO, GSB, DROP, WAIT, FRE, LDIR, LDDR, CALL, BYE, REPEAT—UNTIL—WHILE, RUB, SWAP, FILL, i XATTR.

Verujući da će mnogi naši čitaoci biti zainteresovani za način na koji je Igor Fischer „ubedio" spektrum" da proširi svoj rečnik, zamolili smo pobednika da nam otkrije programerske tajne; taj tekst, kao i razgovor koji je sa prvonagrađenim programerom obavio naš saradnik iz Osijeka, objavljujemo takođe u ovom broju „Računara".

Velika akcija na pozajmljenom računaru

Program „velika akcija" osvojio je drugo mesto na našem Konkursu, a autor ove zanimljive avanturističke igre namenjene na „spektrumu" je Aleksandar Radovanović, student elektrotehnike iz Beograda. To je onaj momak koji vodi školu mašinskog programiranja u „Ventilatoru 202". Pored toga, Aca je i član kluba programera ETF i stručni saradnik „Sveta kompjutera".

Aca Radovanović, sada već iskusni programer, svoje prve kontakte sa računarom imao je još 1982. godine, kada je ZX 81 predstavljao pravu retkost, a od pre dve godine radi isključivo na „spectrumu" kome će, izgleda, ostati veran do kraja.

— Često me pitaju zašto neću da pišem programe za „komodor 64", ili neki drugi računar — kaže Aca. — Pravi dogovor je da me je procesor Z80 razmazio, i da bi programiranje na „komodoru" za mene predstavljalo vraćanje na već prevaziđeno.

Što se tiče ranijih programerskih poduhvata, Aca je, uglavnom, pisao kratke uslužne programe i igre za emitovanje u „Ventilatoru". Tako je i „Velika akcija" prvobitno bila predviđena za „spectrume" vernih slušalaca računarskog dela „Ventilatora", a onda su Acu nagovorili da taj program pošalje na Konkurs.

Aca je u ovoj igri uradio sve potpuno sam — što podrazumeva i izradu scenarija i grafike. Kada je reč o scenariju, tu nije bilo problema; autor je, očigledno, bio više nego inspirisan domaćim ratnim filmovima o mladim ilegalcima. Grafika mu je predstavljala najveću teškoću, jer, kako sam kaže, nikad nije bio naročito vešt u crtanju. Taj „nedostatak", međutim, nije mu smetao da samostalno razvije programe koji su mu omogućili da u samo 7,5 K RAM-a spakuje preko 25 (zapravo 26) slika.

Po Acinom mišljenju, zaplet dobre avanturističke igre sastoji se u rešavanju složenih situacija, a ne u odgonetanju zagonetki kojima se često pribegava da bi igra bila zanimljivija.

— Osim toga, dobra igra-avantura je ona koja nema prevelik rečnik; fond od dvadesetak reči treba da bude dovoljan da se odigra cela igra.

Po tom „receptu" nastala je „Velika akcija", jedna od prvih kompleksnih avanturističkih igara na našem jeziku. Tematika je prilično zanimljiva: „Velika akcija" je blaga i izuzetno uspela parodija na domaće ratne filmove. Ne želeći da otkrivamo tajne igre koja će uskoro biti komercijalizovana,

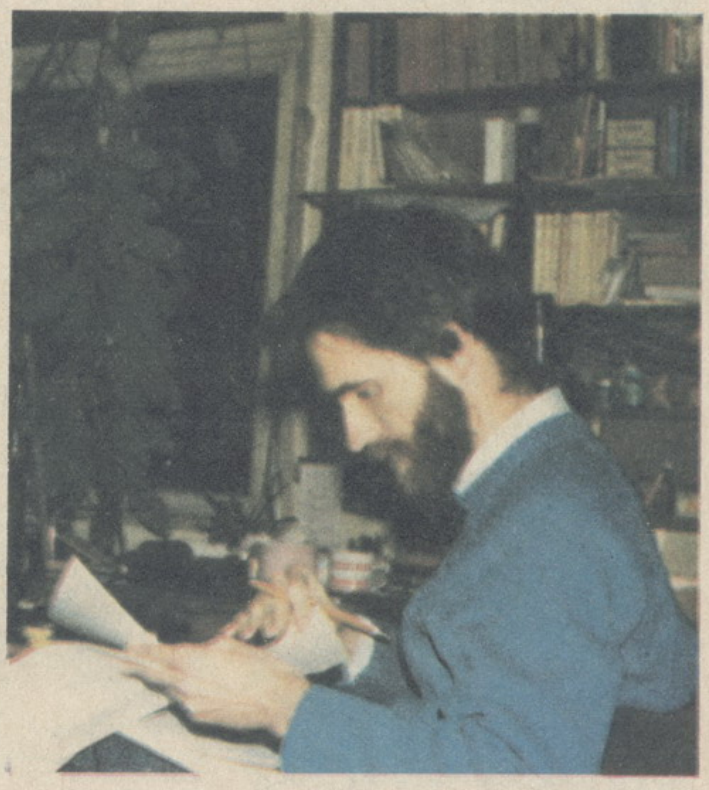
Računar kao deo života

Autor prvonagrađenog programa za kućna računala Igor Fischer, učenik drugog razreda matematičkog usmjerenja Centra za usmjereno obrazovanje „Braća Ribar", stanuje s mamom i tatom u novijoj zgradi u centru Osijeka. Tu između Zimske luke na Dravi i bloka stambenih zgrada nalazi se, na drugom katu, soba tog šesnaestogodišnjaka kojemu je kompjutor već neko vrijeme sastavni dio života. Igrajući se i učeći uz računalo, Igor je uspio svladati nekoliko programskih jezika, proniknuti logiku rada računala, naučiti engleski, pratiti kompjutorsku literaturu, a istovremeno biti odličan učenik, baviti se astronomijom, učiti i fran-

Snimio: Dragutin Olvitz

Igor i računalo: Radni sto s računalom i pomagalima u sobi Igora Fischera

Nakon gotovo jednogodišnjeg nadmetanja za titulu najboljeg YU programera na konkursu časopisa „Galaksija", maratonci su, najzad, stigli na cilj. Sto devetnaest (uglavnom) veoma ozbiljno rađenih programa za petnaestak modela najpopularnijih mašina predstavlja, zaista, bogatu programersku žetvu koja ubedljivije od bilo čega svedoči da računarski podmladak polako izrasta iz kratkih pantalona. Najviše zanatskog umeća pokazali su programeri na „spektrumu", koji su osvojili sve tri glavne nagrade. Prva nagrada dodeljena je Igoru Fisheru, šesnaestogodišnjem učeniku iz Osijeka, koji je svojim radom ostavio iza sebe znatno iskusnije programere, druga Aleksandru Radovanoviću, a treća Slavoljubu Milekiću i Draganu Tanaskovskom. Rezultate konkursa i detaljno obrazloženje žirija objavili smo u aprilskoj „Galaksiji". Ove stranice u „Računarima" posvećujemo portretima najboljih programera.



Kako pojesti engleski: Slavoljub Milekić i Dragan Tanaskovski

čuski, a nedavno je upisao i padobranski tečaj. Prvi skok očekuje u šestom mjesecu.

Nakon trogodišnjeg druženja sa Sinclaireovim računalima, u početku sa ZX 81, sada sa „spectrumom 48K", Igor je postao softveraš koji obećaje. Na to ponajbolje ukazuje i program za proširenje basica „spectruma", što ga je napravio za tri mjeseca i s kojim je pobijedio na ovom natječaju.

— Kad sam imao šest godina, prvi sam puta vidio računalo. Bilo je to veliko računalo u „Saponiji". Tata je tamo radio. Jednog dana nije me imao tko pričuvati u kući, pa me je tata poveo na posao, u računarski centar „Saponije". Moju znatiželju umirio je nekakvim bušačem kartica. Prvi kontakt s kućnim kompjutorom imao sam četiri godine kasnije kod očevog prijatelja. Bio je to „commodore pet". Nakon nekog vremena, imao sam tada 13 godina, započela je u osječkoj Narodnoj tehnici s radom informatička sekcija. Radili smo na „aple"-u. Tu sam započeo učiti šta je to kompjutor — ispričao je Igor Fischer u jednom dahu za vrijeme razgovora u njegovoj sobi. Sjedio je za stolom sa „spectrumom", „grundingom" u boji, kazetofonom i štampačem. Svakodnevni ambijent u kojem živi, uči i radi ovaj, za koji centimetar niži od 180, vitki mladić.

Njegovo prvo računalo

Za Fischerove se može reći da su većim dijelom kompjutorska obitelj. Tata je priznati stručnjak za velike sisteme, predaje na osječkom Ekonomskom fakultetu informatiku, a mama je liječnica u Općoj bolnici Osijek. Zasad se može reći da je sin Igor krenuo očevim stopama, ali na malim kompjutorima.

— Prvo računalo tata mi je kupio 1982. godine. Bio je to ZX81 u kitu. Zajednički smo ga sastavili i, kad je proradio, sjedili smo od jutra do ponoći pred tv ekranom. Tada se mama jako ljutila, prijeteći da će nas najuriti zajedno s računalom. Naučio sam basic i već tada sam počeo raditi u mašinskom jeziku. Početkom 1983. godine tata je kupio „spectrum 48 K". Kasnije sam naučio pascal, asembler, forth.

Kako si učio jezike, brzo?

— Nisam imao većih problema. Na primjer, pascal sam naučio preko vikenda. Uzeo sam kazetu s pascalom za „spectrum" u petak popodne i do ponedjeljka sam već radio u pascalu. Tata je, naravno, bio iznenađen mojim uspjehom.

Da li u školi učiš nešto o računalima?

— U sklopu proizvodno-tehničkog obrazovanja jedan semestar imamo samo nastavu iz informatike. Slušali smo uglavnom informativno za što se kompjutori koriste ili mogu koristiti. Većinom se govorilo o zastarjeloj tehnologiji. Premda je škola opremljena računalima, praktično ću s njima raditi vjerojatno tek u dva završna razreda. Takav je program.

Inače, u školi se okupljamo svake subote, neformalno, na inicijativu profesora Jošnika. U tom našem kompjutorskom klubu kolege demonstriraju svoje nove igre, neko donese novo računalo, programe.

Igor je mladić koji je na određeni način dimenzionirao svoj život. U ljeto 1984. godine bio je na trotjednom tečaju engleskog u Cambridgeu, Velika Britanija. Naravno, nije tom prilikom izostao obilazak Sinclaireovog razvojnog centra. Očekivao je impozantnu zgradu, a vidio je samo tablu s firmom, sve vrlo skromno. To ga se dojmilo.

— Ljeto prije toga dobio sam od jednog

Programer od akcije: Aleksandar Radovanović

reći ćemo samo da je cilj izvesti spektakularnu diverzantsku akciju u okupiranom gradu i prebeći na slobodnu teritoriju. Pri tome se, jasno, upada u silne peripetije koje treba postepeno rešavati, ispoljavajući zavidan stepen hrabrosti i inteligencije i značajno strpljenje. Članovi Kosmisije moraju da priznaju da nisu uspeli da dovedu avanturu do kraja, pa su se u više navrata obraćali autoru za neke indicije, i najzad, za rešenje.

Koji su tehnički kvaliteti programa, uz inventivne ideje, doneli Aci drugu nagradu (Sinclair Spectrum 48 K). Program gotovo do poslednjeg bajta ispunjava 48 K „spectrumove" memorije; pisan je u bejziku uz korišćenje više mašinskih potprograma koji ubrzavaju rad sa grafikom i komunikaciju sa korisnikom. Po startovanju programa, ekran se deli na tri prozora, od kojih donji omogućava unošenje naredbi, srednji prikazuje računareve odgovore, a gornji, poput većine vrhunskih stranih avantura, sliku pejsaža koji zamišljeni ilegalac vidi. Za prepoznavanje naredbi je zadužen bejzik program, što ne znači da će se njegovim listanjem otkriti previše tajni igre. Umesto očekivane gomile IF naredbi, u programu ćemo naći mnogo operacija sa binarnim matricama u koje su upisani svi relevantni podaci.

Inače, Aca Radovanović je na Konkurs poslao još jedan program, akcionu igru pod nazivom „Mica spremačica".

— Moram priznati da sam pomalo iznenađen odlukom Komisije, jer mislim da je „Mica spremačica" bolji program, u koji je uloženo više programerskog truda — kaže Aca. — To je klasična arkadna igra sa tri nivoa, zasnovana na koncepciji Pekmena, i kompletno je rađena u mašincu, dok je „Velika akcija" dobrim delom pisana u bjeziku.

Kompletan Acin „razvojni sistem" sveo se samo na kasetofon i pozajmljeni (!) računar. Zato mu je za „Micu" trebalo nešto više od mesec dana, a za „Veliku akciju", čak, celo leto. Nema sumnje da je druga nagrada otišla u prave ruke; Aca je nepokolebljivi „spektrumovac", a računar će mu stići u pravi čas, jer je upravo započeo rad na novom projektu koji, za sada, drži u tajnosti.

Engleski u devedeset slika

Treća nagrada (računar „galaksija", verzija 8—6) dodeljena je autorima programa „EATENGLISH 1". To su Slavoljub Milekić i Dragan Tanaskovski iz Beograda. Obojica imaju po 29 godina i trenutno su na postdiplomskim studijama. Pored toga, Dragan je i jedan od autora knjige „Spektrum — priručnik".

„EATENGLISH 1" je edukativni program koji pomaže nastavnicima engleskog jezika u praktičnom radu, ali, u principu, ovaj program koji „jede" engleski lako se može pretvoriti u program koji isto tako uspešno jede francuski, nemački, italijanski ili neki drugi strani jezik. Komisija je procenila da se radi o izuzetno dobro koncipiranom i relizovanom programu, koji je nastao u saradnji autora sa profesorima engleskog jezika na Kolarcu i profesionalnim prevodiocima.

— Hteli smo da napravimo edukativni program koji bi ličio na igre — kaže Slavko, jedan od autora EATENGLISH-a.

Da bi bar donekle u tome uspeli, autori su pre otprilike godinu dana započeli rad na ovom programu istraživanjem među školskom decom. Zatim su određeno vreme posvetili analizi kompjuterskih igara, pokušavajući da otkriju šta je to u njima što toliko privlači decu.

— Najveća mana edukativnih programa je što su nezanimljivi. Obično program zahteva da se ide određenim, ustaljenim putem, što ubrzo postaje dosadno — kaže Slavko. — Hteli smo to da izbegnemo, pa smo modularnim prisupom izrade EATENGLISH-a omogućili korisniku da bira način prolaska kroz program."

Kasnije se pokazalo da i to na neki način ograničava korisnika, pa su autori razvili pristup zvani „ALL YOU WISH", čime je program postao „nešto kao korpa sa igračkama iz koje možeš da vadiš šta hoćeš i kad hoćeš".

Rad na ovom programu sastojao se od nekoliko etapa, od kojih je jedna bila pisanje rutina za crtanje: „EATENGLISH" ima preko 90 slika! Program je opremljen rečnikom u kome se nalazi oko 360 parova englesko-srpskohrvatskih reči koje su grupisane u šest blokova: imenice, prevodi, glagoli, opoziti, predlozi i pridevi. Učenje je organizovano preko lekcija koje se učitavaju sa kasete, pri čemu nastavnik ima mogućnost da pripreme nove lekcije i modifikuje postojeće u skladu sa nastavnim planom koji ostvaruje.

Autori su gotovo sve vreme radili odvojeno, tako da se, praktično, nije znalo do poslednjeg časa kako program izgleda u celini. „EATENGLISH" je završen takoreći, u minut od dvanaest, štoga završeni deo posla nije dorađen do kraja. Osim toga, Komisija je procenila da program ima jednu komercijalnu manu koja mu je, možda, oduzela jednu od prve dve nagrade: čini nam se da bi program namenjen samostalnom izučavanju engleskog jezika bio interesantniji za širi krug čitalaca od programa koji je, u osnovi, namenjen profesorima engleskog!

Dejan Ristanović
Jelena Rupnik

Engleza tridesetak igara. Upoznao sam se s njime preko kompjutora. Cijelo ljeto sam sjedio i igrao se. Poslije, kad sam pošao u školu mogao sam hodati ali jedva da sam mogao potrčati — ispričao je Igor objašnjavajući i drugu stranu računalne pasije. Razvija duh, a tijelo?

igra sa jednim životom

Koliko je tata utjecao i razvijao tvoju ljubav prema računalima?

— Tata mi nije posebno pomagao u učenju. Pomogao mi je kupivši opremu, kasnije mi je pomogao savjetima i nabavom literature. Literaturu stvarno sam ne bih mogao nabaviti.

Literatura?

— Najviše koristim „The Complete Spectrum ROM Disassembly". Često čitam, pratim kompjuterske časopise i knjige, bez toga ne bi mogao. Od časopisa kupujem „Galaksiju", „Računare", „Moj mikro" i „Mala računala".

Sada radiš na „spectrumu". Da li on zadovoljava tvoje želje, da li želiš bolji kompjutor?

— Za moje potrebe „spectrum" je dovoljan, ali želje su mi veće. Htio bih malo moćniji kompjutor, s boljim mogućnostima za grafiku i ton. Možda QL, IBM PC, u obzir dolazi i C-64, „commodore 128", „amstrad". Od opreme disk jedinice i pošteni monitor.

Koliko dnevno provodiš vremena uz računalo?

— Ne prođe dan kada ne taknem kompjutor. Nekad ga samo uključim, odigram igru, a nekad sjedim satima za nekim programom.

Razmišljaš li o programiranju igara?

— Razmišljao sa, , pripremao sam se za to, ali sam za sad odustao. Dvojica Osječana napravili su dobru igru, bar mislim da je tako, ali je nisu mogli nikome prodati. Radili su na njoj cijelo ljeto, po dvanaest sati dnevno. Kad sam to vidio, izbubio sam volju. Inače, mislim da je kod tog posla najvažnija ideja. Ostalo je već lakše.

Što poslije srednje škole?

— Elektrotehnika, radi kompjutora, ili možda medicina.

Uspjeh Igora Fischera može biti poticaj svima onima koji su kročili u svijet računala. To nije kompjutorska igra u kojoj se može imati nekoliko života. Uspjeh tu ne dolazi preko noći, ali može doći ako se poneka noć izgubi bulјeći u ekran na kome se smjenjuju listinzi, slova, igre, brojevi, znakovi. Tko u prvom životu ne prođe tu školu, u drugom za to neće imati šanse.

B. Hebrang

dvoboj džinova

Računari u izlogu

Dvaput godišnje, zimi u Las Vegasu i leti u Čikagu, u Americi se održava na daleko čuveni sajam elektronike za široku potrošnju. Po tradiciji, vodeći proizvođači računara predstavljaju upravo na njemu svoje nove proizvode. Tako je bilo i na januarskom sajmu u Las Vegasu — štandovi „Atarija" i „Komodora" bili su pretrpani novim mašinama. Između ove dve firme oduvek je postojala konkurencija, ali se ona naročito zaoštrila kada je nedavno bivši „Komodorov" šef Džek Tremiel poveo sa sobom čitavu ekipu inženjera i preuzeo rukovodstvo nad „Atarijem". Sada obe firme imaju nove mašine i nove — motive za mržnju. Nadamo se da će iz ovog dvoboja džinova ljubitelji računara ipak izaći kao najveći pobednici.

LCD: računar na baterije

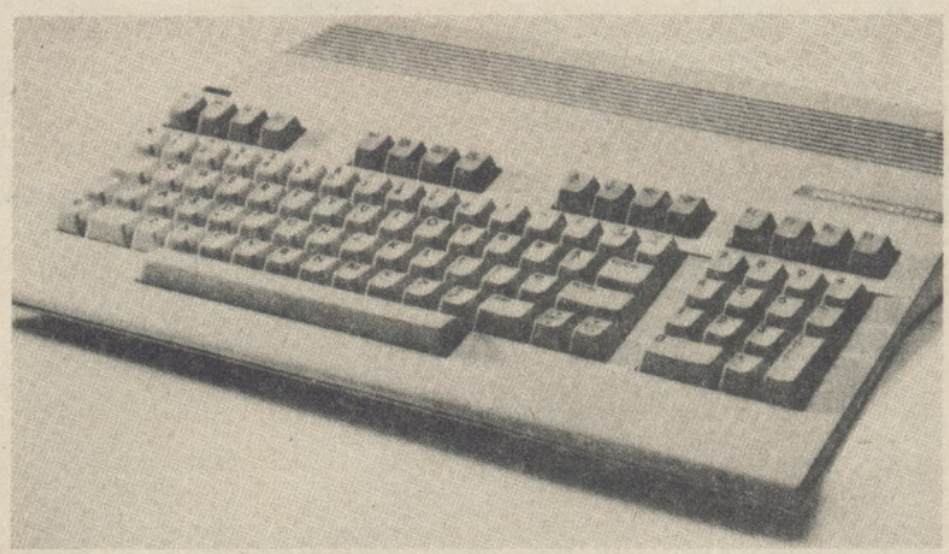
Prvo što se zapaža kod „Komodora LCD" je da izgleda — izvanredno! Računar je smešten u svetlu sivo-žutu kutiju u čijem se poklopcu nalazi displej sa tečnim kristalima. Ispod poklopca je tastatura sa 72 ASCII, 4 kursor i 8 funkcijskih tastera. Da li je to onaj isti „Commodore" koji je dizajnirao ružni CBM-64?

Glavni procesor LCD-a je 65C102. To je proširena CMOS verzija starog 6502 sa novim naredbama. Veoma je sličan ali ne i identičan procesoru 65C02 koji koristi „epl IIc". Vremenska baza procesora je 1MHz, što znači da po brzini ne prevazilazi popularne kućne računare kao što su „epl IIe" i CBM-64. „Komodor LCD" ima 32 K CMOS RAM-a i 96 CMOS ROM-a, koji sadrži operativni sistem i aplikacione programe. CMOS tehnologija troši veoma malo struje — koristeći 4 AA baterije sa LCD-om se može neprekidno raditi 15 časova! Računar se može napajati i preko ispravljača.

U ROM-u se nalazi program za obradu teksta, baza podataka, knjigovodstveni program, adresar, spisak, kalkulator, podsetnik, terminal program, bejzik 3.6 i mašinski monitor. Na računaru postoji port za dodatne ROM-ove. Zahvaljujući ovome, RAM memorija se koristi isključivo za podatke, dok se programi nalaze u ROM-u. Programi se mogu koristiti paralelno. Na primer, ako nam u toku rada sa bazom podataka padne na pamet zakazan sastanak, dovoljno je pretisnuti jedan funkcijski taster, napisati vreme u podsetnik i nastaviti sa radom.

Još jedna od dobrih strana LCD-a je to što ima brojne portove: ugrađen je 300-bodni direktni modem, IEC bas, RS-232C port, centroniks interfejs i čitač bar koda. IEC bas omogućava priključivanje svih CBM-64 periferija, kao što su 1541 disk drajv, MPS-801/2/3 printeri i 1526 priner-/ploter.

Računar ima LCD displej sa 80 kolona u 16 linija, koji se može koristiti kao grafički ekran sa rezolucijom od 480×128 tačaka. Displej je „Commodoreove" sopstvene proizvodnje. Uobičajen problem kod velikih LCD-a je da su teško čitljivi, ali je ovaj, kažu, izvanredan.



Kompatibilnost pre svega: Komodor 128

PC128: tri računara u jednom

Kao i LCD, PC128 izgleda takođe atraktivno — tanka kutija svetlo sivo-žute boje. Računar zauzima više prostora na stolu nego što bi se očekivalo od kućne mašine — 56×432×324mm.

U PC128 su ugrađena tri različita procesora — 6510, 8502 i Z80A — što mu omogućava da radi u tri režima: kao C64, PC128 i CP/M-80. Računar može da „oseti" kada je priključen C64 kartridž ili kada se u drajvu nalazi CP/M disk i automatski se postavlja u odgovarajući mod. Iz jednog moda u drugi se može prelaziti i pod softverskom kontrolom. Da bi se, na primer, prešlo u C64 mod, treba kucati GO ON 64 — računar postavlja pitanje ARE YOU SUPE; Y/N i posle otkucanog Y na 40 kolonskom ekranu se ispisuje bejzik poruka: verzija 2.0 sa 38 K slobodnog RAM-a.

„Commodore" tvrdi da je PC128 u C64 modu hardverski i softverski potpuno kompatibilan sa C64. Tada PC128 koristi 6510 MPU na 1.02MHz, 6581 generator zvuka, 64K RAM-a i 16+4Kb ROM-a. Maksimalna finoća grafike je, kao i kod C64, 320×200 tačaka sa 16 boja i 8 sprajtova. Kompatibilnost ide dotle da se čak mogu koristiti i C64 kartridži. „Commodore" kaže da je isprobao sve postojeće C64 kartridža na PC128 i da su svi radili.

U svom osnovnom modu PC128 koristi 8502 procesor na 2MHz (6502 kompatibilan), 6581 generator zvuka, 128Kb RAM-a i 48+16Kb ROM-a. Operativni sistem, bejzik i mašinski monitor se nalaze u 48Kb ROM-a, dok je u preostalih 16K smeštena podrška diska. RAM može biti proširen do 512K, ali je 128K najveća količina sistemske memorije. Ostatak se može koristiti iz mašinskih programa ili kao RAM disk.

U najvišoj grafičkoj rezoluciji ekran je podeljen na 640×200 tačaka, a u srednjoj na 320×200. U grafičkom modu od 320×200 tačaka mogu se koristiti 16 boja i 8 sprajtova preko superprijateljskog bejzika V7.0. Za korisnika bejzik 7.0 ostavlja slobodno 122 K (tačnije, 122365 bajtova). To je proširena verzija „Commodoreovih" bejzika 2.0+3.5+4.5 sa ukupno 140 naredbi. On podržava grafiku i zvuk i omogućava struktuirano programiranje DO . .LOOP . .WHILE . .UNITIL . .NEXT.

Pored ovoga, PC128 mog omogućava brzi prenos podataka koristeći „Commodoreov" novi disk drajv 1571. U C64 modu brzina transfera podataka je ista kao između CBM-64 i 1541 drajva, dok se u ostalim modovima brzina višestruko povećava.

Ugrađen Z80A CPU, a samim tim i mogućnost korišćenja CP/M operativnog sitema bez ikakvih proširenja računara, predstavlja deo težnje „Commodora" ka standardnim sistemima.

U ovom modU PC128 je standardna 8-bitna CP/M mašina. Koristi CP/M Plus 3.0 koji se automatski učitava sa disk i omogućava pristup svih 128K RAM-a. Dodatni RAM se može koristiti kao RAM disk.

CP/M koristi Z80 na 4MHz pri čemu se ispis vrši na 40/80 kolonskom ekranu u 16 boja.

Ulazno/izlazni portovi su u punom „komodore" standardu — korisnički port, port za kasetofon, serijski (IEC), port za proširenja, dva upravljača porta za police za igre, svetlosnu ulovku i miša. Tu su, takođe, i kompozitni video, RGB i TV izlazi, tako da računar nije teško priključiti na bilo koji monitor ili TV.

Tastatura računara podržava razne modove. Na prvi pogled, radi se o profesionalnoj izvedbi (92 tastera) sa posebnim grupama od 14 numeričkih i 8 programskih funkcijskih tastera. Ako se pažljivije zagleda, primećuju se dve grupe tastera za pomeranje kursora — jedna na gornjem, a druga na donjem delu glavne grupe tastera. Ovo omogućava kako elektronsku tako i potpunu komptibilnost tastature sa CBM-64.

U CP/M režimu PC128 radi podjednako dobro kao i u C64 modu, što mu otvara vrata u dve najveće biblioteke programa. Postavlja se pitanje koliko će se programa pisati za „prirodni" PC128? Pored sto-dvadeset-osmice, predstavljen je i PC128D. Kod PC-a 128 u jednoj kutiji se nalazi „sva pamet, i ugrađeni 1571 disk drajv, dok je tastatura spojena kablom.

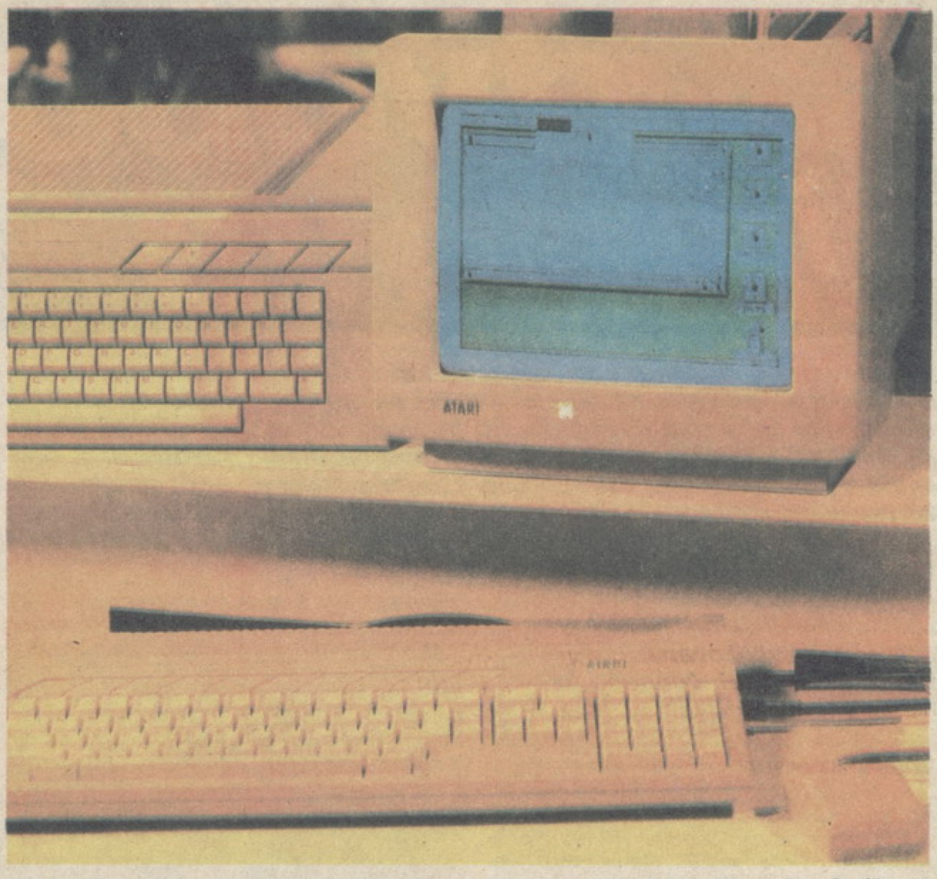
Atarijeva „snaga bez cene"

Još čod svojih prvih (davnih) uspeha „Atari" nije napravio ništa vredno pažnje. Sve vreme dok je bio pod upravom „Warner Brothers" nije radio ništa osim što je gubio pare. „Warner Brothers" je tada odlučio — da proda „Atari". Na scenu tada stupa elegantno popunjeni Džek Tramiel (Jack Tramiel), dotadašnji stegonoša u „Komodoru". On i njegov klan imali su reputaciju teških ljudi i njihov dolazak u „Atari" bio je propraćen nezadovoljstvom, jednog dela kompanije. „Atari" je sa Tramielom, međutim dobio i veliki broj „Komodorovih" inženjera. Nema sumnje da je njegov dolazak transformisao „Atari": započet je širok razvojni program pod parolom 'snaga bez cene'. Ovaj program obuhvata sve — od 8-bitnih kućnih do 32-bitnih poslovnih računara. Za sve ove mašine karakteristična je izuzetno niska (po nekima samoubilačka) cena.

„Atari" je izabrao januarski CES da prikaže prve nove proizvode: 4 8-bitna modela (poboljšane verzije 800XL) i 2 nova 16/32-bitna računara sagrađena oko MC68000.

Četiri nova '65' modela bazirana su na uspehu 800XL. Atari tvrdi da su sve šezdesetpetice kompatibilne sa serijom 400/800 uključujući i 1050 disk drajv.

Osnovni model '65' serije je 65XE. Izgrađen oko procesora sa 64K RAM-a, novajlija, kao i svi „Atarijevi" računari, ima dobru grafiku i zvuk, koje kontrolišu posebno dizajnirani čipovi za te svrhe.



Novi „atari": Rat cenama do istrebljenja

Grafičke mogućnosti 65XE uključujući i grafičkih modova sa maksimalnom rezolucijom od 320×192 tačaka, 5 modova za tekst i tradicionalnu „Atarijevu" 'player-misile' sprajt grafiku.

Spolja, 65XE je sasvim različit od 800XL. Kutija je tamno sivo/bele boje kao kod starih računara. Funkcijski tasteri koji su se nalazili na desnoj strani tastature pomereni su iznad glavne grupe tastera.

Ostale mašine serije 65 slične su modelu 65XE, samo nude specijalne prednosti. Tako 65XE poseduje mnogo šire zvučne mogućnosti nego 65XE: umesto 4 osnovna glasa, XEM ima haj-faj zvuk u 8 glasova sa 64 harmonika.

Model 65XEP je portabl verzija 65XE. Pored celokupnog hardvera 65XE, XEP sadrži ugrađeni 5 inčni monohromatski monitor i 3.5 inčni „Sonyjev" mikroflopi drajv. Sve ovo se nalazi u jednoj kutiji. Poslednji model serije '65' je 130XE. Uprkos svom nazivu '130', on je identičan po svemu sa 65XE sem po tome što ima 128K RAM-a.

Cene za računare '65' serije još nisu utvrđene. Sve što je „Atari" mogao da kaže bilo je da će biti najmanje koliko i cena B00XL, a to je $120.

Modeli 130ST i 520ST su prva dva „Atarijeva" računara nove generacije moćnih ličnih računara. U osnovi, to su iste mašine — jedina razlika je što 130ST ima 'samo' 128K RAM-a, a 520—512K.

Mozak ST-a je MC68000 — jedan od najbržih mikroprocesora koji se danas mogu nabaviti. Ovaj MPU se koristi kako u manjim (ali izuzetno moćnim) računarima, kao što je „mek", tako i u nekim super-mini sistemima.

U osnovnoj verziji ST, ima kompozitni video, RGB i TV izlaze. Za video memoriju koristi se 32K RAM memorije. Grafika je izvanredna. Računar ima paletu od 512 boja i radi u 3 grafička moda: 320×200 sa 16 boja, 640×200 — 4 boje i 640×400 u dve boje.

Muzika se dobija preko 3 nezavisna glasa i MIDI sintesajzer interfejsa. Postojanje interfejsa omogućava korišćenje velikog broja popularnih sintesajzera. Ovo je jedini kućni (poslovni?) računar koji se u osnovnoj verziji dobija sa ovim interfejsom.

Kao spoljašnja memorija mogu se koristiti disk jedinica od 3,5 inča i hard diskovi. Računar poseduje disk kontroler, hard disk interfejs i mogućnost DMA — direktnog pristupa memoriji bez učešća procesora. Pored ovih, računar ima RS232 serijski i centroniks paralelni interfejs, kao i dva porta za djojstik. Jedan port je napravljen tako da može da radi i sa „mišom" koji se dobija uz računar.

Jedna od najzanimljivijih prednosti ST računara je sistem pod imenom GEM (Digital Research) proširen specijalno za „Atarijev" operativni sistem nazvan TOS — Tramiel Operating System. Ovaj sistem, kao i operativni sistem Apple-ovog „meka", omogućava istovremeni prikaz više programa na ekranu pomoću takozvanih prozora, kao i korišćenje „miša". GEM se upotrebljava na mnogim računarima, ali je ST prvi koji ga ima u izvornoj verziji.

Jedini problem u svemu ovome je softver. „Džekintoš" (kako mnogi nazivaju ST) nije kompatibilan ni sa jednom postojećom mašinom. Mada „Atari" tvrdi da mnoge softverske kuće pišu za ST, u to ne treba preterano verovati.

Očekuje se da će 130ST koštati oko $300, a 520 ST $500. Ovo je prvi računar na MC68000 koji košta manje od $1000.

Predrag Bogdanović

Šesnaestogodišnjaci za računarom

mladi genijalci ili izgubljena generacija

Računari u razgovoru

Dr Jozo Dujmović, vanredni profesor za računarsku tehniku i informatiku

Nema, izgleda, te stvari koja može toliko da pokrene YU naciju kao fudbal i — računari. Računarsko opismenjavanje postaje, odista, „formula jedinstva" oko čijih skuta se okupljaju svi — od desetogodišnjaka i penzionera koji na instant kursevima priučenih programera po mesnim zajednicama gutaju osnove bejzika, preko domova kultura, klubova, omladinskih organizacija, najpopularnijih radio i televizijskih emisija i najčitanijih dnevnih, nedeljnih i mesečnih novina, gomile specijalizovanih časopisa, do škole i školskih vlasti. Računarski jezik postaje važniji od bilo kog stranog jezika. Cela se nacija njiše u ritmu bejzika sa „spektruma" i „komodora". Dr Jozo Dujmović, vanredni profesor za računarsku tehniku i informatiku na beogradskom Elektrotehničkom fakultetu, uputio je nedavno (polu)otvoreno pismo prosvetnim vlastima u kome ustaje protiv računarske euforije i postojećih koncepcija računarskog obrazovanja. Šta je našeg uglednog sagovornika — čoveka koji za sobom ima sedamdesetak naučnih i stučnih radova iz oblasti računarstva, čiji je projekat za izbor „Data Base Managment" sistema pobedio u velikoj konkurenciji na konkursu koji je za potrebe agencije američke vlade raspisao National Bureau of Standards iz Vašingtona, jedno vreme vanrednog profesora za oblast „Computer Science" na univerzitetu države Florida — navelo da javno uzme u odbranu računare od onih koji ih, reklo bi se, najviše vole?

Profesore, obrazovanjem u programiranju bavite se još od 1967. godine, kada ste prvi put izabrani za asistenta. Vas svakako nije iznenadila računarska groznica koja je, mada sa kašnjenjem, ipak stigla do nas, ali vam, kako ste naveli u pismu upućenom na više adresa institucija zaduženih za obrazovanje, smeta način kako se kod nas rešava problem računarskog opismenjavanja i obrazovanja. Krenimo od početka. Kako doći do osnovne računarske pismenosti?

Do osnovne pismenosti se dolazi u osnovnoj školi. Tu se sadi seme koje će davati plodove tokom celokupnog daljeg školovanja pojedinca. Zbog toga je osnovno računarsko školovanje stvar veoma delikatna i o njoj treba krajnje ozbiljno porazmisliti. Tu smo, za sada, na samom početku, ali ima indicija da i taj početak preti da krene pogrešnim putem.

Kako po vašem mišljenju, treba pristupiti podučavanju programiranja u osnovnim i srednjim školama?

Ima nekoliko bitnih elemenata kojih se treba pridržavati. Prvo, programiranje treba, naročito u osnovnim školama, da se shvati kao disciplina koja se suštinski oslanja na nastavu matematike. To znači da je bitno podvući one elemente u programiranju koji predstavljaju algoritamsko rešavanje problema. Da bi se došlo do dobrih algoritamskih rešenja za razne probleme, potrebne su dve komponente: mogućnost definicije i struktuiranja desetak osnovnih tipova podataka, i mogućnost primene desetak dobro struktuiranih kontrolnih mehanizama. Očigledno, ove koncepcije se ne bi smele uvoditi na primeru bejzika, jer u većini slučajeva on ima samo dva tipa podataka (brojni i znakovni) i samo tri osnovne kontrolne strukture (IF, FOR i ON).

Računari kao ugrožena vrsta: Dr Jozo Dujmović

Ovo su elementarne i nepobitne činjenice. Na žalost, veletrgovci bejzikom, koji su u potpunosti zagospodarili ne samo masovnim medijima već i našim žalosno zbunjenim prosvetnim vlastima, ne uvažavaju činjenice. Razloge za to možete lako da pogodite i sami. Naravno, masovni mediji imaju prvenstveni cilj da zarade novac zabavljajući mase, a to znači kompjuteramatere i vlasnike kućnih računara, pa je tu razumljiva potreba za bejzikom i računarskim mađioničarstvom koje uz njega ide. Međutim, kada su prosvetne vlasti u pitanju, tu opraštanja ne može da bude. Na kraju krajeva, oni se spremaju da svojim „bejzikogalaktičkim" idejama nevešto eksperimentišu i na mojoj deci.

Stvarni ili pseudojezici

Pa šta onda treba da bude prvi jezik programiranja koji se uči u bilo kakvoj školi računarstva i programiranja?

Tu su stvari odavno jasne i mogu se proveriti u svim dobrim knjigama posvećenim projektovanju i analizi računarskih algoritama. Prvi jezik treba da bude jedan matematički korektno odabrani PSEUDOJEZIK koji ne opterećuje nepotrebnim tehnikalijama vezanim za konkretni računar i konkretni jezik (izbegava se potreba za uvodom u operativni sistem, editore teksta, prevodioce, tastature, internu organizaciju

„Veletrgovci bejzikom su potpuno zagospodarili ne samo masovnim medijima već i našim žalosno zbunjenim prosvetnim vlastima"

računara i ostalo). Cilj pseudojezika je da direktno i bez ikakvih ograničenja uvede KONCEPCIJE programiranja, koristeći svo bogastvo struktuiranja podataka i kontrolnih mehanizama. Takav pseudojezik redovno podseća na moderne algoritamske jezike kao što su paskal, ada, modula, C, fortran 77 i (u izvesnoj meri) PL/1. Prema tome, algoritamske probleme treba FORMULISATI i REŠAVATI na pseudojeziku, a zatim se rešenja mogu sa lakoćom PREVESTI na bilo koji konkretan jezik, uključujući (u pomanjkanju boljeg) i bejzik. Tek pošto su udareni jaki temelji pravilnog algoritamskog razmišljanja, može se pristupiti konkretnom računaru i konkretnom jeziku. Tada se konkretan programski jezik doživljava kao SPECIJALAN SLUČAJ onoga što je poznato iz pseudojezika. Naravno, tako se izbegava da se nanese šteta do koje bi došlo ako bi se siromaštvo jednog bejzika ili jedne „galaksije" podiglo do nivoa opšte teorije.

Koji realni jezik bi bio najbolji za prvi kontakt dece sa računarima?

Tu ne postoji idealno rešenje za sva vremena. U sadašnjem trenutku i sa sredstvima kojima raspolažemo to bi mogao da bude paskal. Kao što znate, pascal radi na računarima koji imaju oko 48 kilobajta operativne memorije i ne moraju imati diskete. Takve mašine koštaju manje od 100 američkih dolara. Verujem da su nama dostupne (t.j. mogu se lako razviti i masovno proizvoditi) i nešto skuplje školske mašine. Tamo gde nema ništa drugo sem bejzika, treba koristiti bejzik.

Drugi jezik koji treba obavezno držati na oku je mikro-prolog. Ovaj jezik ima nekoliko izvanrednih osobina. Razvijen je primerno sa ciljem da se približi školskoj deci, ima mogućnosti da radi sa bilo kakvim objektima i relacijama, uvodi koncepcije neproceduralnog programiranja i rada sa bazama podataka. To je, bez sumnje, jedan od jezika budućnosti.

Žaba i princeza

Pomenuvši „bejzikogalaktičare", dodirnuli ste dva problema — izbor programskog jezika i izbor računara. Kako vidim, ne odobravate ni to što su prosvetni organi izabrali da se naš prvenac koristi u osnovnim školama. Svakako, bolje je da se obuka iz programiranja vrši na računaru velikih mogućnosti sa solidnom programskom podrškom, ali možete li vi da nam navedete neki konkretan domaći mikroračunar koji ne košta stotine miliona, a mogao bi se razviti u specijalizovan školski računar?

Dobar čarobnjak može od žabe da napravi princezu. Kod nas ima puno žaba i malo čarobnjaka. Ako treba da konkretno odgovorim, onda moj izbor pada na PMP koji je u Institutu Jozef Stefan u Ljubljani razvio ing Marjan Miletić. Ta mašina je čarobna po tome što predstavlja legendarni PDP-11 po ceni od samo 250 dolara. Kada se ta mašina opremi sa vezom za televizor u boji, pretvoriće se u prvorazredan školski računar. Obilje softvera pod operativnim sistemom RT-11 odavno postoji.

To je više od šezdeset hiljada dinara već u startu. Kada se doda veza za TV u boji, dobijamo cenu proizvodnje koja je bar 100 000 dinara, a koliko bi tek taj računar koštao u prodaji?!!! Mi smo siromašno društvo. Već šest generacija učenika matematičko-tehničke struke obučavamo za programere i operatere dakle, ne dajemo im osnovnu računarsku pismenost već ih profesionalno obrazujemo, a odgovarajuće škole u Beogradu nisu od SIZ-ova usmerenog obrazovanja dobile sredstva čak ni za nabavku „galaksija". Imamo, doduše, osam akademija nauka, 18 univerziteta, stotine SIZ-ova, ali SREDSTAVA NEMAMO, a s obzirom na našu ekonomsku situaciju, ne verujem ni da ćemo ih uskoro imati. Budimo realni, može li se napraviti još jeftiniji računar, a da opet bude na raspolaganju „obilje softvera"?

Sigurno da može. Najbolji primer za to je Sinclair „spektrum", na kome su raspoloživi paskal, C, prolog, ligp, logo, nekoliko bejzika i nekoliko asemblera. To je računar sa kojim pokušava da se takmiči „galaksija" i slične mašine, ali bez uspeha.

Smatrate li da deci u prvom kontaktu sa računarstvom treba da se izlažu koncepcije unutrašnje organizacije računara, mašinskog jezika i asemblera, kako je zamišljeno u sadašnjim nastavnim programima?

Ne — iz više razloga. Jedan od njih je i mogućnost da „fon-nojmanovske" arhitekture računara budu u najskorijoj budućnosti zamenjene sasvim drugačijim koncepcijama baziranim na paralelnom procesiranju.

Kada se govori o računarskoj pismenosti, pitanje je da li je uopšte nužno da svi uče programiranje?

Razume se da nije nužno. Ono što svi treba da znaju to je ZAŠTO i KAKO treba primeniti računar da se olakša posao i svakodnevni život. Pri tome ne mislim samo na mikroračunare već na računare uopšte. Kao što je nezamislivo da civilizovan čovek ne ume da se posluži telefonom, jednako je neophodno da bude u stanju da se koristi i računarom. Kućni računar povezan sa telefonom i običnim televizorom daje fantastične mogućnosti i bez programiranja. Stoga se važnija komponenta računarske pismenosti može dobiti iz kursa pod naslovom „Računari i društvo" (koji se na Zapadu redovno predaje) nego iz kurseva programiranja.

Izbeglice u računarstvo

Govorili ste o tome kako se kod nas fabrikuju laici. Na koji način onda stvaramo stručnjake?

Računarska tehnika i informatika je profesija kao i svaka druga. To podrazumeva da se inženjeri računarske tehnike moraju sistematski pripremati kao i svi ostali regularno školovani stručnjaci, t.j. na fakultetima za računarstvo i informatiku. Takvi

„Precenjujemo ulogu amatera, glorifikujemo samoučki prilaz programiranju, ne uočavajući opasnost koja deci preti od pogrešno naučenih metoda"

fakulteti (na Zapadu poznati kao „Computer Science") postoje skoro na svim modernim univerzitetima. Na žalost, na Beogradskom univerzitetu za sada ne postoji fakultet za računarstvo i informatiku. Smerovi za računarstvo „gostuju" na nekim fakultetima, na primer na Elektrotehničkom fakultetu, na Prirodno-matematičkom, i na FON-u. Međutim, u svakom od navedenih slučajeva radi se o relativno malom broju redovnih studenata, koji se moraju otimati od drugih disciplina koje se na tim fakultetima takođe predaju. Fond časova nije dovoljan i privre-

„Ta svest se rađa u momentu kada dojučerašnji pirat napiše svoj prvi program u nameri da ga prodaje i ubrzo postane i sam žrtva pirata"

da sigurno ne dobija sve one kadrove koji su joj u domenu računarstva potrebni. Tako dolazimo do fenomena da je broj studenata koji upsuje postdiplomske studije iz računarstva i informatike veći od broja redovnih studenata, što pokazuje da privreda jasno traži kadrove, ali Univerzitet za te potrebe nema sluha. Ovo opet ima i tu posledicu da se u domenu računarstva kreću mnogi polustručnjaci-izbeglice iz raznih drugih profesija, neregularno školovani, često samo u školama koje drže trgovci računarima, t.j. razni predstavnici u svojim školskim centrima pokušavaju ne da školuju profesionalce, već da pripreme svoje kupce za elementarno korišćenje računara.

Beogradski univerzitet, glomazan i rascepkan, slabo finansiran i u domenu računarstva sasvim zapostavljen, ostavlja da se stvari rešavaju same od sebe po fakultetima, onako i onoliko koliko naporima pojedinaca može da se postigne. Društvo, u ovom slučaju Republika, pa i grad Beograd, ne pokazuju ni interes, ni organizovanost, pa ni najobičnije razumevanje za potrebe šireg regiona u domenu visoko obrazovanih kadrova za računarstvo i informatiku. Najviše zabrinjava neverovatna činjenica da ne postoje efikasne društvene institucije koje bi u SR Srbiji bile zadužene za brigu za razvoj računarstva i informatike.

Ono što ispušta društvo pokušavaju, izgleda, da nadoknade kompjuterske novine. Kako sredstva javnog informisanja prikazuju računarsko opismenjavanje kod nas?

Na žalost, većina novinskih članaka ne uspeva da se oslobodi euforije i senzacionalističkog prikazivanja naših prvih koraka u omasovljavanju korišćenja malih računara. Tako se precenjuje ili pogrešno tumači uloga kompjuter-amatera, glorifikuje se samoučki prilaz programiranju računara, ne uočava se opasnost koja deci preti od pogrešno naučenih metoda programiranja, amatersko programiranje se interpretira kao ozbiljna privredna delatnost, a privatni osnivači „softver kompanija" se prikazuju kao privrednici koji su otkrili novu šansu za međunarodnu afirmaciju jugoslavenske privrede.

Neka se deca uče

Šta vam najviše smeta u napisima o računarima?

Posebno me iritira što se na softverski vandalizam i softverski šund gleda kao na stvari koje su dobre i progresivne. Na primer, u NIN-u od 28. oktobra 1984. možete pročitati sledeće reči pune nacionalnog ponosa: „nema čak ni najskupljih američkih sistemskih programa koji ovde nisu 'provaljeni' i kopirani", i dalje, „naši klinci, koji moraju da razbiju zaštitu programa da bi ga izmenili ili presnimili, postaju tako u softverskom smislu pismeni". Šta na osnovu ovoga mogu o nama misliti oni koji su napisali te provaljene i iskopirane sistemske programe, a da ne pominjemo one čiji je posao da te programe održavaju, prodaju i pravno štite? Druga izjava je biser posebne vrste: ona indicira da čovek može da postane pismen onda kada ukrade knjigu u biblioteci ili na sajmu knjiga i ne vidi očigledan paralelizam da je presnimavanje (a kasnije i prodaja presnimljenog programa) isto što i presnimavanje gramofonske ploče i prodaja tako dobijenih kaseta. Naravno, kada je reč o gramofonskim pločama svi će se složiti da se radi o kriminalu (to, uostalom, piše ne samo na svakoj ploči, već i na početku svakog programa) i većina ljudi će gramofonsku ploču ili kasetu kupiti u dućanu a ne od privatnog muzičkog pirata. Na žalost, kada se pogledaju oglasi u novinama (ili na radiju) vidi se da ono što izgleda normalno za snimke muzike nije normalno i u slučaju računarskih programa: rezultati tuđeg rada se nemislosrdno „provaljuju", kopiraju, i zatim bestidno oglašavaju za prodaju. Ono što nas ovde zabrinjava nije što neki vešt dečak presnimi program i razmeni ga sa svojim drugom, već što je odsustvo javnog morala i dobrog ukusa dospelo dotle da se na organizovano razbojništvo u softveru gleda sa razumevanjem i simpatijom kao na znak „računarske pismenosti".

„Gubimo vreme u naivnoj veri da će samouka računarska omladina na prokrijumčarenim kućnim računarima preporoditi računarstvo u Jugoslaviji"

„Razvikani sedamnaestogodišnjak koji završava svoj treći model kućnog računara nije 'konstruktor računara' nego ekvivalent radio-amateru iz pedesetih godina koji hrabro sklapa prijemnik sa pet radio-lampi"

Dobro su poznati slučajevi „hekera" koji su koristeći lične računare i telefonsku mrežu prodrli u datoteke na računarima velikih američkih kompanija ili vladinih agencija i mnogi su i kod nas i u svetu na to gledali kao na dobar „štos", a ne kao na čist kriminal. Koliko ovo može da iritira prave profesionalce vidi se i po tome što je Ken Tompson, koautor proslavljenog operativnog sistema UNIX, primajući nedavno Tjuringovu nagradu (najveće priznanje za rad u oblasti računarske tehnike) posvetio dobar deo svog izlaganja borbi protiv računarskog vandalizma i kriminala. Po Tompsonu, provaljivanje u tuđi softver ili provaljivanje u podatke na tuđem računaru isto je što i provaljivanje u tuđi stan i preturanje po tuđim fiokama i ormanima. Pri tome nema olakšavajuće okolnosti u činjenici da je stan možda bio otključan. Dok takvi postupci nisu zakonski kažnjivi oni su u domenu vandalizma, a kada uđu u krivične zakone (na čemu se u mnogim zemljama upravo intenzivno radi), onda ulaze u domen teško kažnjivog kriminala, jer ugrožavaju ne samo pošten rad i privređivanje, već često i bezbednost društva. Naravno, u većini slučajeva računarska omladina nije u potpunosti svesna čime se, u stvari, bavi kada nekažnjeno presnimava i prodaje računarske programe. Ta svest se rađa u momentu kada dojučerašnji pirat napiše svoj prvi program u nameri da ga prodaje i ubrzo postane i sam žrtva ostalih pirata. To je uočio i pomenuti NIN-ov članak: kada mi krademo i presnimavamo tuđe pismo, na to se gledalo sa simpatijom kao na izraz zdravog duha, preduzimljivosti i želje za računarskim opismenjavanjem, a kad nas isti ti naši simpatični razbojnici pokradu, onda se najednom ustanovi da nešto nije u redu i to se naziva „softverski paradoks"! Nema tu nikakvog paradoksa, već se prosto radi o logičnoj posledici naopakog ponašanja u kome se nikakva autorska prava ne poštuju ni od strane pojedinca, ni od strane društva. Najstrašnije je, možda, to što se nedavno takva delatnost posebno rasplamsala pod okriljem beogradskog Doma omladine koji je ponosno organizovao sajam ukrdenog softvera!!

Šta je onda vaša poruka „klincima koji trguju tuđim softverom"?

U šali bih rekao neka gledaju „Gangstersku hroniku", pa neka zaključe koji su im dalji koraci u karijeri. Ozbiljnije, moja poruka računarskoj omladini je jasna: čak i kada nije kažnjivo, neovlašćeno provaljivanje, preturanje, i korišćenje tuđih programa je vandalizam sličan čitanju i zloupotrebi tuđih pisama, a neovlašćeno stvaranje dobiti na tuđem radu je kriminal. U krajnjoj liniji na sve ovo treba gledati sa stanovišta morala, ne radeći nikada ono što ne bi želeli da drugi rade vama.

Kafanski svirači na računaru

U opštoj programerskoj euforiji čak i najtrivijalniji program nekog novog klinca dočekuje se ovacijama. Šta podrazumevate pod softverskim šundom?

Računarska pismenost slična je pismenosti u običnom smislu. Ako je neko pismen, to ni u kom slučaju nije garancija da će ono što napiše biti automatski veliko literarno delo. Ko god je video šund u novinskim kioscima, lako će zamisliti kako izgleda softverski šund. Osnovna osobina softverskog šunda, slično kao i kod literarnog šunda, je da se vrlo dobro prodaje. To je osnovni motor koji pokreće pisce mnogih računarskih igara i osnivače softverskih kompanija. Naivni novinari nas obaveštavaju da „program daje računaru smisao i to se sjajno plaća: od dve do šest hiljada funti za dva-tri meseca rada" i zatim u tome vidi da je „softver naša jedina šansa". Ne pitaju se, pri tome, kakav je to program, kakav je to smisao, i kakva je to šansa! Pa valjda je blisko pameti da bi naša devizno pregladnela privreda odavno otkrila tu šansu da su stvari tako proste kako se to novinarima na prvi pogled pričinjava. Očigledno je da postoji šansa da pojedinac dobro proda

„Dosta softvera koji proizvode mladi samouci željni brze slave i još brže zarade spada u kategoriju softverskog šunda"

program koji uz obilje svetla i zvuka tamani horde marsovaca fotonskim torpedima, ali je to daleko od „naše jedine šanse" i sigurno nije delatnost na koju se može računati kao na siguran putokaz do privrednog preporoda. Dosta softvera koji proizvode mladi samouci željni brze slave i još brže zarade, naročito kada su u pitanju razne računarske igre, spada, na žalost, u kategoriju softverskog šunda i upravo to je oblast do koje se najpre dolazi pošto se pojedinac elementarno računarski opismeni. Proizvodnja softvera može da bude ozbiljna privredna delatnost, ali taj softver (složeni aplikacioni sistemi, sistemi sa bazama podataka, programski prevodioci, operativni sistemi, i slično) zahteva ozbiljnu pripremu i stručno znanje koje daleko prevazilazi mogućnosti dece koja se igraju računarima.

Svakako znate da je u Americi masovni prodor računara u svakodnevni život počeo upravo zahvaljujući amaterima. Iza prvog ličnog računara „alter" i najbolje prodavanog računara svih vremena „epla" stoje amateri, možda genijalni, ali amateri koji nemaju naročito visoko školsko obrazovanje. Da li su ono što zovete „kompjuter-amateri" predznak napretka ili možda izgubljena generacija?

Programiranje računara je veština koja podseća na veštinu sviranja violine: najpre, treba imati sluha, zatim treba solidno, dugotrajno, i strpljivo učiti osnove tehnike sviranja, i konačno, treba svakodnevnim sviranjem održavati stečenu tehniku. Kako violinu sviraju samouci? Na žalost, po pravilu loše, i čak i kada su talentovani na kraju ne postanu više nego kafanski svirači. Sa porogramiranjem je slična situacija. Deca koja do računarskog programiranja dolaze preko samoučkog čitanja polupismeno napisanih programa za razne računarske igre, ili preko amaterskog kursa bejzika, nisu na pragu velikog otkrića najvažnijeg jezika 21. veka — suprotno tome, oni su u opasnosti da postanu izgubljena računarska generacija koju će 21. vek iznenaditi koncepcijama veoma različitim od bejzika i od današnjih dečijih računara. Mnogi primeri iz školske prakse razvijenih zemalja ukazuju da studenti koji uče moderne koncepcije računarske tehnike i programiranja postižu bolje rezultate ukoliko im ne smeta znanje samoučki stečeno na kućnim i sličnim računarima.

Kompjuter-amateri su u potpunosti slični radio-amaterima. Očigledno je svima da poštansku telekomunikacionu mrežu ne mogu da prave radio-amateri već profesionalni inženjeri telekomunikacija. Po istom principu, razvikani sedamnaestogodišnjak koji završava treći model kućnog računara nije „konstruktor računara" već ekvivalent radio-amateru iz pedesetih godina, koji je najpre napravio detektor, pa zatim dvocevnik, i sada hrabro sklapa prijemnik sa pet radio-lampi.

Računarstvo bez motike

Imamo li, posle ovoliko kašnjenja i propusta, uopšte neku realnu pravu „računarsku" šansu?

Ne postoji kratka, prosta i magična formula koja će otkriti našu „jedinstvenu šansu" u oblasti računarske tehnike i informatike. Naša prava šansa je u tome da se što pre opametimo i shvatimo da kod računara, kao i kod ostalih privrednih delatnosti, nema hleba bez motike, nema velikih rezultata bez velike pripreme, nema mesta amaterizmu u modernoj tehnici. Deca koja se igraju računarima neće razviti računarstvo u Jugoslaviji — većini će igra dosaditi pre ili posle. Računarstvo mogu da ozbiljno razviju samo ozbiljno pripremljeni profesionalci — pod uslovom da ih uopšte budemo imali. Da bismo takve ljude proizveli, moramo biti mudriji na celom putu od osnovne škole pa do doktorata. U osnovnoj školi moramo razviti kvalitetnu nastavu osnova računarstva i računarske matematike, polazeći od principa koje nalažu računarske nauke, a ne od trenutno raspoloživih računara-igračaka i od raspoloživih ali, na žalost, prevaziđenih softverskih koncepcija. U domenu srednjeg školstva treba mnogo uložiti da bi stvorili solidan stručni kadar koji će u oblasti poslovne informatike moći da radi poslove koje danas obavljaju priučene fakultetlije. U domenu visokog obrazovanja krajnje je vreme da se oformi lanac fakulteta računarske tehnike po uzoru na „Computer Science" fakultete na zapadnim i istočnim univerzitetima. Računari nisu više skupi kao što su bili pre 20 godina i ništa se korisno neće postići inkvizitorskim sprečavanjem uvoza računarske opreme i moderne literature — svaki dolar koji država na ovaj način danas uštedi pretvoriće se u sto dolara gubitka u bliskoj budućnosti.

Ako na svemu ovome budemo brzo, smišljeno i organizovano radili, možda ima nade da u ponečemu i uspemo, a ako prepustimo stvari da se „same organizuju" i da vreme prolazi u naivnoj veri da će samouka računarska omladina na prokrijumčarenim kućnim računarima preporoditi računarstvo u Jugoslaviji, onda su nam neuspeh i dalje zaostajanje u potpunosti zagarantovani. Ima još jedna stvar koja je savršeno jasna; računarstvo i informatika su oblast o kojoj treba da brine država — ona treba da prva shvati značaj problema i da ga rešava njoj dostpunim sredstvima, i to na svim nivoima istovremeno. Ne bude li toga . . . ode i ovaj voz.

* * *

Nadamo se da je ovo tek jedan od prvih ozbiljnih razgovora vođenih na temu računarstva i obrazovanja iz računarstva u našoj zemlji. Pošto smo nekako preživeli, po zvaničnim stavovima „uspešnu reformu školstva", treba da počnemo sa novom koju spremaju isti oni ljudi koji su kreirali i prethodnu, ali su u međuvremenu znatno promenili mišljenje o tome koliko i kakvih nam kadrova treba. Pokušaćemo da im pomognemo koliko možemo da pacijent posle ove druge ne umre, jer verujemo da imaju još puno dobrih ideja.

Razgovor vodila Nevenka Spalević

rados

Većina kućnih računara je projektovana sa dobrim grafičkim mogućnostima, pre svega zbog korišćenja za video igre. Neiscrpna mašta i slikarski talenat programera ponudili su vlasnicima mikroračunara igre sa izuzetno lepom grafikom. Ako i sami dobro poznajete programiranje, možete sopstvene kreacije da sačuvate na disku ili traci, izdate ih na štampaču ili ploteru i da ih kombinujete u raznim programima. Međutim, većina programera koji bi želeli da koriste sopstvene crteže ne poznaje toliko dobro programiranje, a i da zna, ovakvo crtanje ne pruža mogućnost komfornog rada, kao što je, na primer, lako dodavanje ili brisanje linija.

Elektronsko platno

Ono što je potrebno za jednostavan rad je ulazni uređaj koji može da prevede linije i krivine sa vaše skice na papiru u informaciju koju bi vaš računar razumeo i mogao da prenese na ekran. Taj uređaj, koji na tržištu profesionalnih računara postoji gotovo isto tako dugo koliko i sami računari, zove se grafička podloga (graphic pad), grafička tabla (graphic tablet) ili grafička ploča (graphic board). Nedavno su se pojavile jeftinije verzije pristupačne vlasnicima kućnih računara. Na njima se crta pomoću specijalne pisaljke (stylus), s tim što se crtež može pratiti na ekranu. Čak i najprostije grafičke table omogućavaju da crtež obojite nekom od boja koje računar može da proizvede ili da koristite poteze „četkicama" različite debljine.

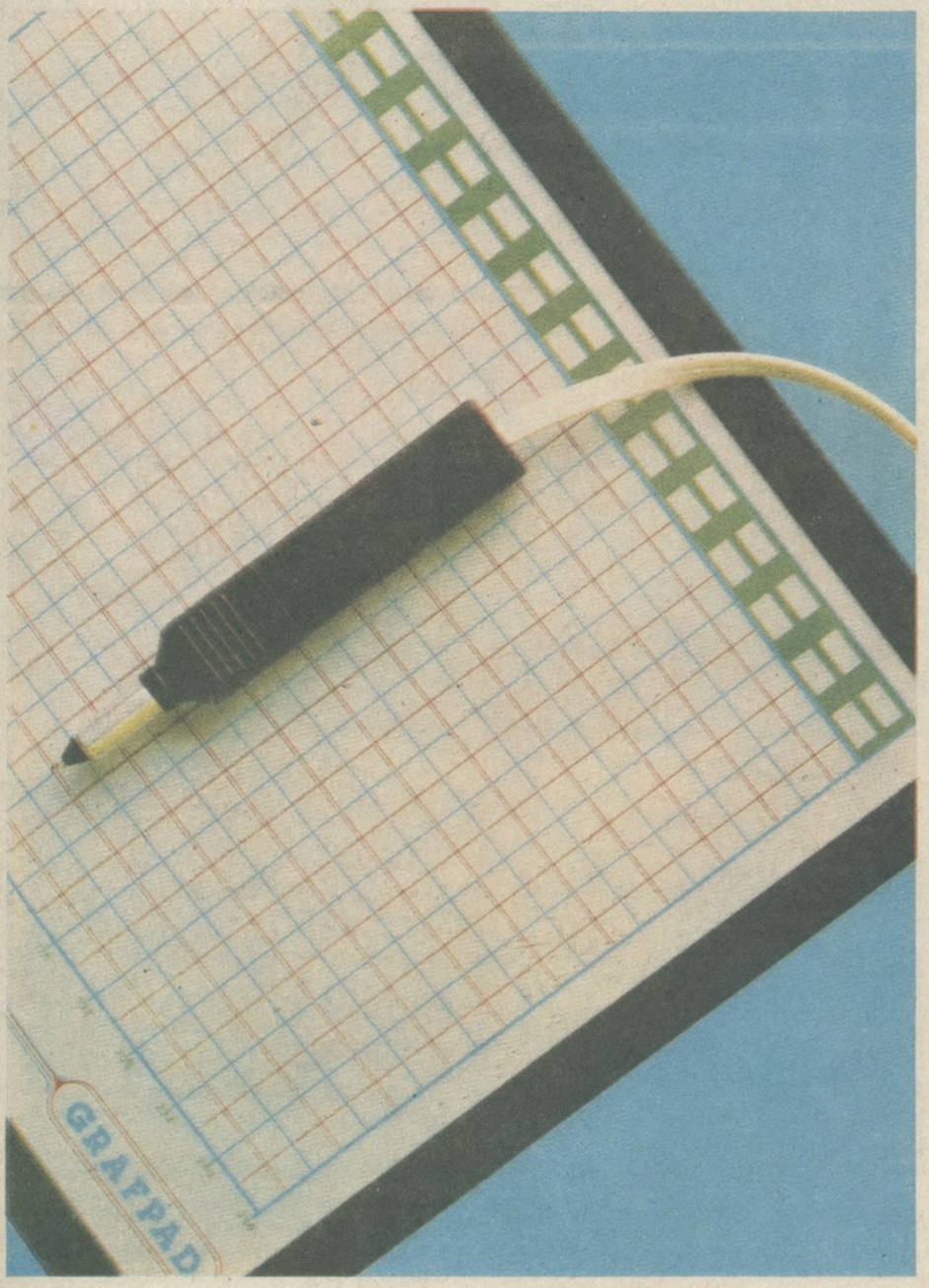
Većina grafičkih tabli i prateći programi nude slične mogućnosti crtanja. Razlike nastaju u manipulaciji crtežom. Na primer, neke grafičke table imaju tzv. „zum" koja omogućava uvećanje nekog dela crteža i popravljanje njegovih detalja obično do nivoa tačke (piksel). Dobre grafičke table omogućavaju pomeranje nekog dela crteža po ekranu, ponavljanje nekog elementa slike više puta, izmenu mesta detalja sa slike ili raznih slika i automatski efekat ogledala. Crtež se obično može sačuvati na traci ili disketi, ali razne grafičke table u kombinaciji sa raznim računarima memorišu ekran na različite načine. U nekim kombinacijama sačuvana slika može se koristiti u drugim programima, ali ovo ne važi za sve table.

Većina kućnih računara može biti isprogramirana za „skidanje" slike sa ekrana na štampač (dump). Ova pogodnost trebalo bi da je predviđena i za grafičku tablu ako vam je u radu potreban ovaj oblik kopije. Mada za sada nije moguće dobiti kvalitetnu višebojnu štampu, boje se mogu sugerisati različitim stilovima osenčavanja. Najbolji rezultati postižu se korišćenjem plotera, ali su oni veoma skupi, a proizvođači vašeg računara ili grafičke table možda nisu predvideli njihovo povezivanje sa ploterom.

GRAFIČKE TABLE

t crtanja

Jedna od oblasti u kojoj su računari prevazišli sva očekivanja je i oblast dizajna. Tehnički crtači, arhitekte, dizajneri i svi oni koji crtaju za svoju dušu imaju uz računar neslućene olakšice u radu. Računari svih veličina, od velikih do kućnih, koriste se danas za sve vrste projektovanja i crtanja.



Skuplja od računara za koji je namenjena: Grafička tabla „Grafpad" za računar „spektrum" ima velike izražajne mogućnosti, ali joj mašina kojoj je namenjena nije baš sasvim primerena

Brojevi i slike

Problem digitalizacije često se rešava u računarima, jer je mnoštvo primena vezano za rad sa analognim veličinama. Signal se u analognom sistemu menja kontinualno i može uzeti bilo koju vrednost iz određenog intervala. Klasični primeri rada sa analognim signalima su kontrole pritiska, temperature ili jačine zvuka. Računar koji treba da vrši ove kontrole razume jedino diskretan digitalni (cifarski) signal. Svi njegovi elementi mogu se naći jedino u dva stabilna stanja, 0 i 1, i stoga su sve međuvrednosti za računar nerazumljive.

Digitalizacija je proces pretvaranja kontinualnih signala koji mogu značiti da su neki elementi „poluisključeni" ili „sivi" u niz diskretnih signala koji će opisati odgovarajuće stanje jedino pomoću vrednosti „uključeno" i „isključeno" ili „crno" i „belo" razumljive za računar. Sklopovi koji vrše ovo pretvaranje zovu se A/D pretvarači ili konvertori (analogno digitalni konvertori). U računarima se, takođe, koriste i konvertori koji vrše pretvaranje digitalnih u analogne signale — D/A konvertori. Oni su sastavni delovi onih izlaznih uređaja koji treba da daju analognu informaciju, sliku ili zvuk, na primer.

Ulazni uređaj po imenu grafička tabla pretvara sliku u brojeve koje preuzima računar. Ove brojeve izlazni uređaj ekran opet pretvara u sliku i mi bez zalaženja u suštinu pretvaranja informacije sa ulaza možemo da pratimo svoje poteze pisaljkom. Međutim, bez metode pretvaranja slike u brojeve bilo bi nemoguće da računar radi sa crtežima i ogromno polje grafički orijentisanog rada računara bilo bi nemoguće.

Proces digitalizacije slike sastoji se od deobe slike u što je moguće više jednakih delova. Što je veći broj delova (obično kvadratića) veća je i sposobnost beleženja detalja. Kod velikih komercijalnih računara rezolucija je dovoljno velika, pa je njihova slika tek neznatno zrnaste strukture. Grafičke table za mikroračunare rade na istim principima, ali slika ide u manje detalja. Kada bismo nacrtali linije preko slike da dobijemo predstavu kako je ona podeljena u kvadratiće, izgledalo bi kao da je slika prekrivena providnom milimetarskom hartijom ili mrežom kao na geografskim kartama. Neke table kao „Grafpad" zaista su podeljene na taj način. Svaki od tih kvadratića ima svoje koordinate po x i y osi koje predstavljaju razumljivu informaciju za računar i omogućavaju mu da vodi evidenciju o svakoj poziciji. Većina grafičkih tabli je projektovana tako da im se rezolucija poklapa sa rezolucijom računarovog displeja, što znači da je broj kvadratića na tabli isti kao broj tačaka na ekranu.

Najveće razlike između tabli leže u načinu proizvodnje signala. Neke zahtevaju specijalnu pisaljku i proizvode signal na osnovu kontakta pisaljke i aktivne zone grafičke table, dok je za druge dovoljan dodir pisaljke ili, čak, pritisak prsta. Među najpreciznijim metodama su magnetski i kapacitivni sistemi. Kapacitivni sistemi nude tačnost do 0,02 mm — dovoljno za inženjere i tehničke crtače — a magnetski sitemi su potencijalno još bolji. Oba tipa koriste žičanu mrežu ugrađenu u tablu. U magnetskim sistemima žičana mreža se

koristi za otkrivanje pozicije pisaljke koja je, u ovom slučaju, nemotaj koji emituje impulsni visokofrekventni signal. Signal koji detektuje mreža upoređuje se sa jednim referentnim signalom i daje neposredno x i y koordinate pisaljke. U ovoj grupi je „Grafpad" koji se može koristiti sa računarima „spektrum" „BBC i komodor 64".

Kapacitivni sistemi rade na suprotan način, jer se pisaljka koristi za otkrivanje niza kodiranih impulsa koji se uvode u dvoslojnu mrežicu.

Softver za crtanje

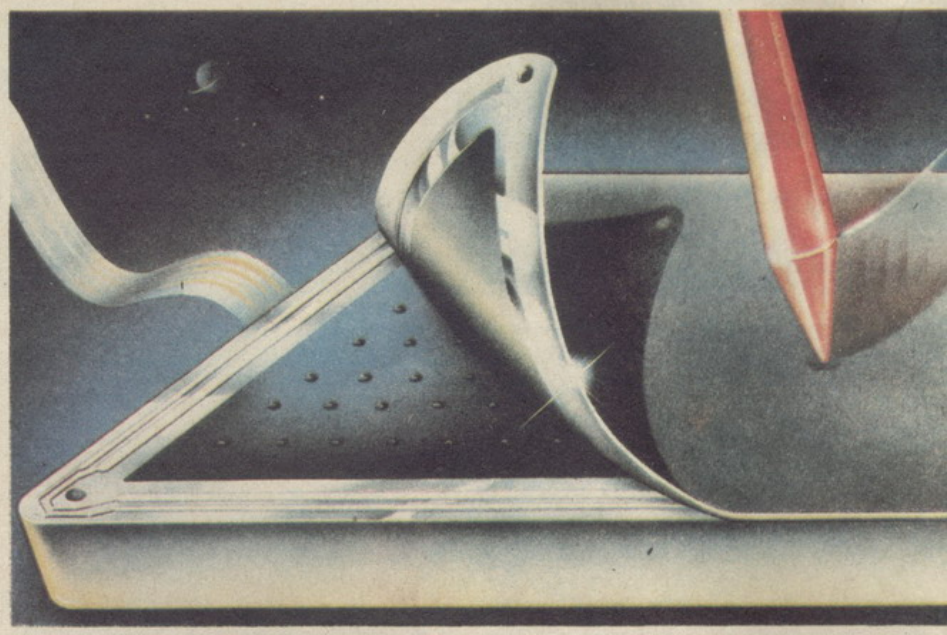
Bez obzira na očigledne hardverske razlike, komunikaciju grafičke table i računara omogućuje programska podrška. Ona ne definiše samo raspoložive osobine već i lakoću korišćenja table, pa pri izboru treba, pre svega, voditi računa o pratećem softveru. Do jednog nivoa, kvalitet sistema zavisi od vašeg računara. Što je računar brži i što ima veću memoriju, više je i prostora za upotrebu naprednije grafičke table. Naravno, programska podrška treba da koristi sve pogodnosti ugrađene u hardver vaše mašine. Karakteristika dobre grafičke table je da vam na pristupačan način stavi na raspolaganje sve mogućnosti računara.

Softver za crtanje kod grafičke table vrlo je sličan programskim paketima za dizajn koji omogućava kontrolu tastature, s tim što tabla može biti mnogo zgodnije programirana od tastature. Zajedničko za sve grafičke table je da omogućavaju da se oponaša prirodna akcija crtanja. Pre početka kreiranja slike treba iz ponuđenog menija izabrati način rada. Izbori menija mogu se vršiti pritiskom na taster ili korišćenjem same table. U tom slučaju, ponude su postavljene na ekranu, tako da, recimo, vidite red mastionica, red četkica i razne druge opcije. Izbor se vrši pomeranjem pisaljke kojom se vodi kursor na ekranu do prave pozicije, a potvrđuje pritiskom na dugme.

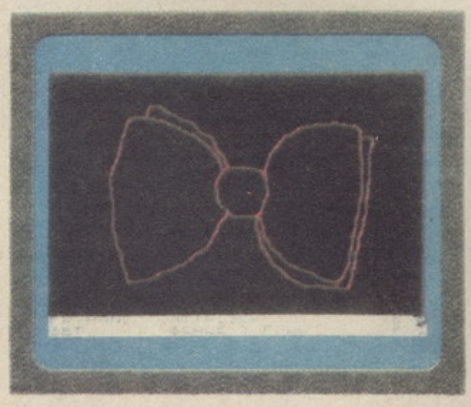
Crtanje na tabli može biti slobodnom rukom ili preko postojeće slike koju možete fiskirati na ploči. U ovoj i ostalim aplikacijama glavni posao softvera je da interpretira signale sa ploče i da sopšti računaru gde treba da ih prikaže na ekranu. Tako se oblik koji crtate odmah prikazuje na displeju. Slobodno crtanje može biti iznenađujuće teško ako, na primer, želite da povučete pravu liniju. Najlakši metod kreiranja regularnih konstrukcija je korišćenje ugrađenih geometrijskih oblika — pravih linija, krugova, pravougaonika i sl. Oni se biraju iz menija, a zatim se pisaljka koristi za fiksiranje položaja i veličine oblika početne i završne tačke.

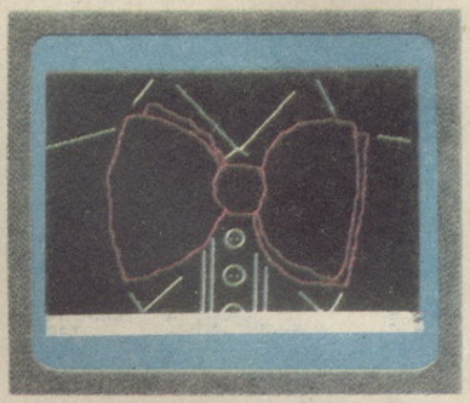
U bilo kojoj tački procedure obično postoji mogućnost brisanja i prepravke slike, mada je ovo ograničeno na poslednje operacije izabrane sa menija, jer računar može da drži samo prethodni ekran u memoriji. Greške koje smo uočili prekasno za ovu vrstu korekcije mogu se nekad izbrisati nadcrtavanjem u boji pozadine, ali ovo ne mora uvek da bude efikasno. U krajnjem slučaju, možete izbrisati sve i početi iznova.

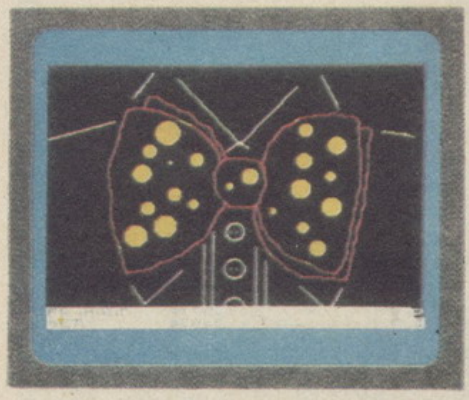
Boja crteža može, po potrebi, biti promenjena tokom crtanja. Postoji, takođe i mogućnost bojenja oblika. Taj proces se obično naziva „crtaj i popunjavaj". Prvo se

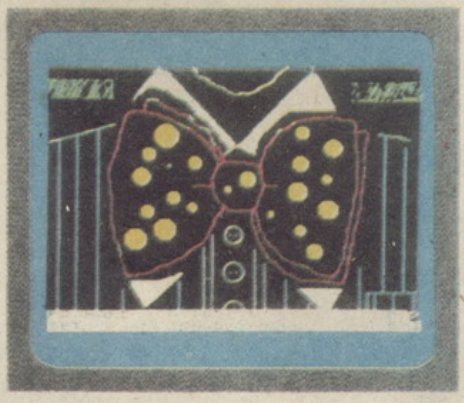


Moćno oruđe u spretnim rukama: Grafičke table značajno pojednostavljuju rad na dizajn





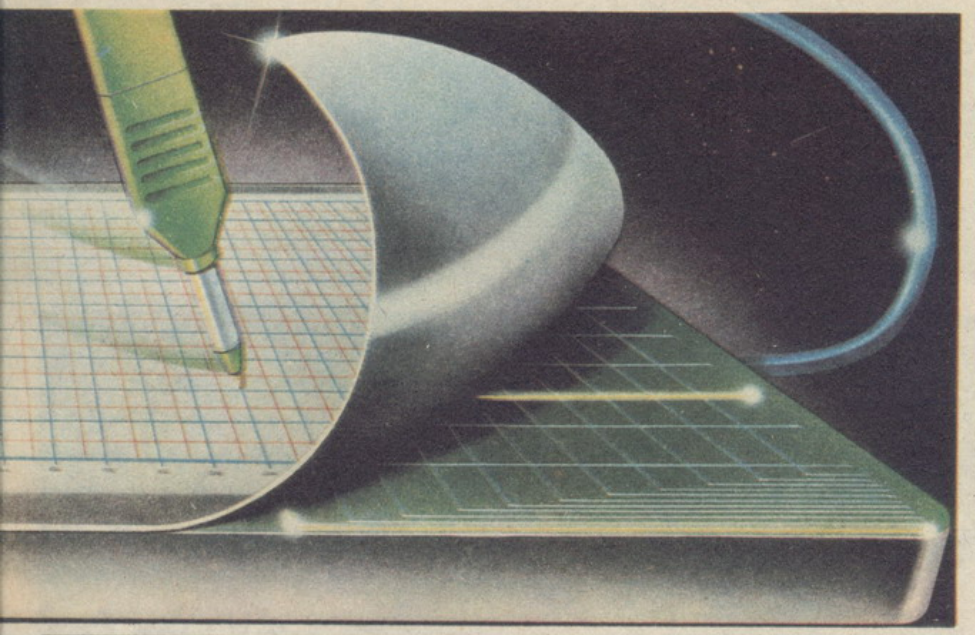




Mašna u četiri slike: Crtež na grafičkoj tabli nije ni iz daleka tako grub kako bi se moglo pomisliti

crta spoljašnja linija, a onda se unutrašnjost oboji postavljanjem kursora u nju i izborom odgovarajuće boje. Iz oblika koji nisu potpuno zatvoreni boja će „curiti" svuda naokolo. Specijalizovane opcije variraju mnogo više od ploče do ploče. Primeri tih opcija su mogućnost ogledanja oblika (ogledalski efekt) i kopiranje oblika. Tako, na primer, možete da nacrtate celu šumu kreiranjem samo jednog drveta, ogledalo se koristi za alternativni oblik, a zatim vrši kopiranje. Postoji i mogućnost uvećavanja dela ekrana, što, kombinovano sa opcijom crtanja, daje mogućnost da do najsitnijih detalja uredite sliku. Neke table omogućavaju i „tačkasto" crtanje (stippling efekt).

Kad završite sliku, možda ćete poželei da je sačuvate. Softver za crtanje pruža i ovu mogućnost, uz ograničenja vezana za način na koji računar memoriše sliku (Spectrum i BBC, na primer, čuvaju ceo ekran, dok CBM-64 u multikolor modu čuva ekran u tri dela). Svi sistemi omogućavaju čuvanje slike na traci i disku. Ovako memorisane slike mogu se ponovo pozvati na doradu, a moguće je i kombinovanje ele-



Koliko košta

„Grafpad"	„spektrum", BBC	£143,75	British Micro
„Koala pad"	„komodor 64"	£ 80,00	Audiogenic
„Animation Station"	„komodor 64"	£ 80,00	Consumer Electronics

Korisne adrese
British Micro, Unit Q2, Penfold Works, Imperial Way, Watford, Herts, WD2 4YY

Slike iz „Koale": Osnovni meni za izbor režima rada i dve opcije — povećanje detalja i umnožavanje pojedinih elemenata slike

menata sa raznih slika. Neke programske podrške nude na tabli set imidža koje možete da koristite za vaše slike. Ako želite da sliku stalno imate pri ruci, treba vam kolor ploter. Alternativa je da fotografišete ekran, vodeći računa da zamračite spoljašnje osvetljenje koje bi moglo da izazove neželjene odsjaje.

Raspoložive vrste

Postoji više vrsta grafičkih tabli za kućne računare. Kompjuterski časopisi prepuni su reklama za ove uređaje, počev od „Micropada" i „Penpada" za IBM PC, koji, pored ostalog, omogućavaju prepoznavanje slova pisanih običnom olovkom i njihovo reprodukovanje u standardnoj formi, do popularnih modela, kao što su „Grafpad" za BBC Micro, koji je nedavno adaptiran i za rad sa „komodorom" i „spectrumom", „Koala pad" za CBM-64 i „Touchmaster" za „dragon".

Kod nas ovaj relativno nov ulazni uređaj još nije izborio svoje mesto među ljubiteljima računara. Za one koji i pored visoke cene i zabrane uvoza odluče da pojačaju svoj računar grafičkom tablom, navodimo neke detalje iz napisa objavljenih u „Your Spectrumu" i „Commodore User-u" o grafičkim tablama za ova dva kod nas najpopularnija računara.

Henri Badžet (Henry Budgett) u svom tekstu za spektrumovce najviše pažnje posvećuje „Grafpadu". Pored ostalog, on kaže da se problemi mogu očekivati već kod povezivanja table sa „spektrumom" ali ako sa uspehom prođete ovu prepreku i učitate program za podršku crtanja koji se dobija uz tablu, možete da krenete u avanturu otkrivanja mogućnosti ovog uređaja — avanturu jer uputstvo (postava od fotokopirnog A5 papira — prema prikazu) nije jaka strana ovog kompleta.

Da bi korisnik imao uvid u ono što se trenutno dešava na ekranu izdvojena je svetla linija na kojoj su zapisane informacije o tekućem režimu rada, izabranoj boji i slične. Za aktiviranje pisaljke treba pritisnuti šiljak tako da mikroprekidač napravi kontakt i onda kursor na ekranu prati pokrete pisaljke po tabli. Za korišćenje bilo koje izabrane funkcije treba pritisnuti ENTER dirku i funkcija će raditi sve dok je pritisnut šiljak. Njegovo oslobađanje završava funkciju.

Ne treba odmah očekivati mnogo, jer crtanje slobodnom rukom zahteva praksu, a i ako je imate, slike nisu potpuno precizne. Srećom, programska podrška ovo toleriše i omogućava da se, pošto se postigne aproksimativna sličnost, editovanjem slika sredi. Najkorisnija među editorskim olakšicama je MAGNIFY, koja povećava izabrano područje do 8 puta. Pojedinačne tačke su oko veličine karaktera i mogu se postaviti i ukloniti po želji, Pokret pisaljke je takođe skaliran, pa slika ne „skače". Vredne pomena su i LINE, CIRCLE i BOX rutine, koje su elastično vezane za bilo koju početnu tačku. Uz ovo, možete još izvrnuti neki lik, pomerati ga gore-dole i levo-desno, i smestiti do dve slike bilo negde u memoriju bilo na traku. Ako želite da zaštitite zonu vaše slike od oštećenja postoji, WINDOW opcija koja dozvoljava da se jedino zona u „prozorčiću" menja dok je opcija aktivna.

No bez obzira na sve ove olakšice, po mišljenju Henri Badžeta, za „spectrum" je „Grafpad" suviše skupa investicija jer se i sa dobrim ekranskim editorima, koji koštaju znatno manje, mogu postići slični rezultati.

Za „komodore 64" na tržištu ima više grafičkih tabli. Pomenimo najpopularniju: „Koala pad", „Grafpad", „Power pad" i „Animation Station". Uz odgovarajući softver — a samo serija „Leonardo's Library" nudi 14 različitih programskih paketa za „Power pad" — grafičku tablu možete koristiti u vrlo različitim oblastima, kao što su matematika, muzika ili crtanje. Tako, recimo, uz muzički program „Micro Maestro" i tablu na kojoj je prikazana notna lestvica i klavijatura, jednostavnim dodirom neke note ili dirke možete da svirate ili komponujete. Ceo proces može se pratiti na ekranu. Zahvaljujući multisenzorskoj tehnologiji, koja može da oseti odgovori i na više dodira u istom trenutku, imate mogućnosti da kreirate čitavu harmoniju. „Micro Maestro" ne zahteva da striktno koristite 4/4 takt, pamti pauze i sinkope i svira upravo onako kako vi komponujete.

Jedan od prvih softverskih paketa proizvedenih za „Koala pad" je „Koala Painter" koji pruža niz mogućnosti — od dodele i promene boja crtežima do kopiranja, kombinovanja detalja sa raznih slika i zumiranja.

Na žalost, i pored mnoštva softverskih paketa za podršku grafičkih tabli CBM računara, do sada nije bilo značajnih koraka za njihovu standardizaciju.

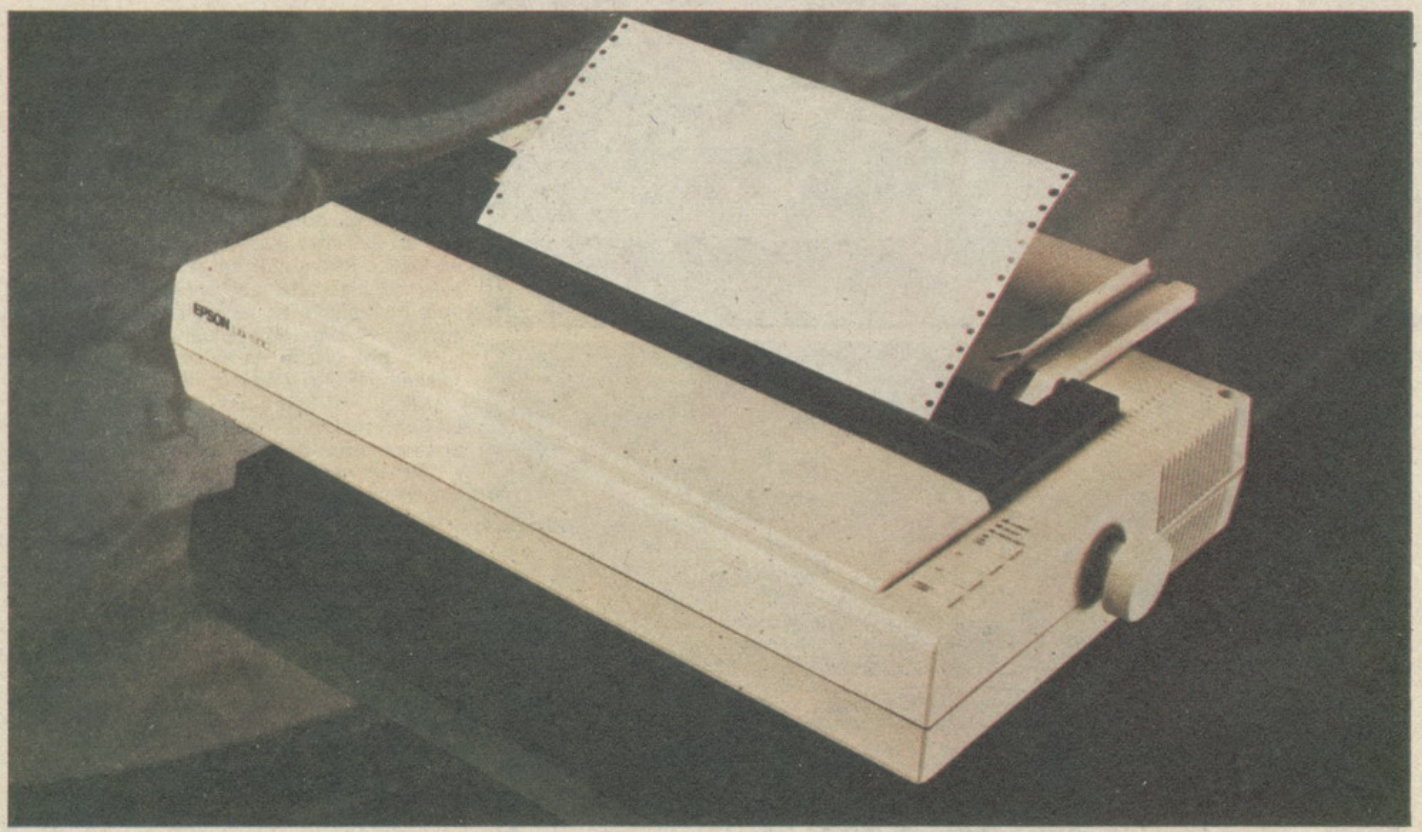
Za grafičke table postoji mnoštvo primena koje se u ovom trenutku ne mogu ni sagledati. Tako se one mogu, na primer, iskoristiti za postavljanje teksta u program, za obradu teksta, za dizajniranje tabele spredšita, kreiranje mustre za tapete, kroja za odeću ili, pak, projektovanje čitavih zgrada. Spomenimo i specijalne efekte na filmu za koje se sve više koriste računari (setite se „Rata zvezda" i „Trona"). Čini se da će kompjuterska grafika, zahvaljujući upravo ovakvim uređajima koje može da koristi širok krug vlasnika računara brzo i značajno izmeniti mnoge oblasti ljudske delatnosti.

Vladimir Ilijevski
Nevenka Spalević

neki novi „epsoni"

Periferijska oprema

EPSON LQ — 1500



Novi standard za matrične štampače: Epson LQ-1500

Šta je to što jedan matrični štampač mora da pruži da bi se svrstao u najvišu kategoriju? U prvom redu — solidnom brzinom postići KVALITET ISPISA BLIZAK KVALITETU SLOVA SA PISAĆE MAŠINE. Tako dolazimo do novog pojma: LETTER QUALITY (otuda ono LQ u nazivu štampača) koga ćemo uslovno prevesti kao „slovni kvalitet". Slovni kvalitet je mogućnost matričnog štampača da generiše karaktere čiji je kvalitet otiska blizak kvalitetu otiska na pisaćoj mašini ili štampaču sa lepezom. Ne upuštajući se u raspravu tipa „matrični ili štampač sa lepezom", jugoslovenski vlasnici računara će biti „osuđeni" na matrične štampače sve dok neko u svetu ne počne da proizvodi YU lepeze jer se samo na matričnom štampaču mogu generisati ćirilica i YU latinica.

Pitanje se samo nameće: šta je to što nekim matričnim štampačima omogućuje, a nekima ne, da postignu taj kvalitet blizak kvalitetu slova na mašini za pisanje. Odgovor je kratak ali i dosta „skup": to je nešto kompleksniji oblik glave za štampanje. Sa njom ide i neophodna složenija elektronika za kontrolu iglica u glavi. Otuda i viša cena ovih štampača. Ali to je već pravilo: ono što je kvalitetno mora da bude i nešto skuplje.

LQ-1500 ima glavu za štampanje (PRINTHEAD) sa 24 iglice — najveći broj koji se obično koristi. I odmah mala digresija: šta, u suštini, predstavlja broj pinova tj. „tačkica" na glavi za štampanje.

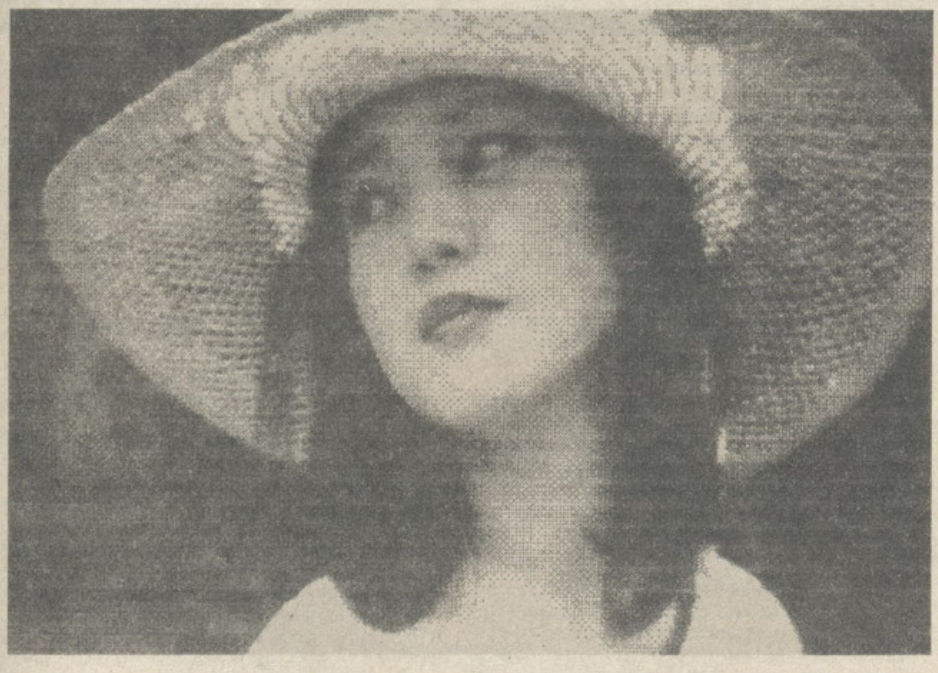
Znamo da je donja granica matrice za kreiranje karaktera 5×7. Međutim, znak dobijen ovakvom matricom je stvarno skromnog kvaliteta, da ne kažemo lošeg, jer ne postoji mogućnost da se deo slova spusti ispod donje linije (slova p, q, y, itd.). Sa matricom koja ima 9 tačaka po vertikali to je već moguće i ispis koji se dobija je sa stanovišta čitljivosti zadovoljavajući — to je onaj poznati računarski izgled teksta. Danas su u upotrebi glave za štampanje sa 9 pinova, 18 i 24. Kod glave sa 18, pinovi su raspoređeni u 2 reda koji su pomaknuti za pola tačke jedan u odnosu na drugi, tako da, u suštini, po vertikali raspolažemo sa 18 tačaka. Glava za štampanje sa 24 pina (kao kod LQ-1500) ima tri reda sa po 8 tačaka takođe sa pomakom redova za pola tačke što ukupno daje 17 tačaka po vertikali.

Dva su osnovna pristupa kod generisanja visokog slovnog kvaliteta: može se koristiti jednostavna glava sa jednim redom od 9 pinova i u više navrata odštampati jedan znak ili se može koristiti složenija glava (sa 18 ili 24 pina) i raditi u jednom prolazu. Odmah zaključujemo da je vreme potrebno za štampanje jednog znaka u slovnom kvalitetu sa deveto-pinskog glavom znatno duže nego kod štampača sa glavom od, recimo, 24 pina. Dobar kvalitet kod prvih se postiže sa čak 4 pristupa

SPECIFIKACIJA

STAMPAC :	LQ-1500
PROIZVOĐAC :	EPSON AMERICA, Inc., 3415 Kashiwa St., Torrance, CA 90505
DIMENZIJE :	60cm x 37cm x 13cm
TEZINA :	14 kgr
OSOBINE :	24-pinska glava za štampanje, brzina 200 kar./s za standardni oblik karaktera, 67 kar./s za tzv. LETTER QUALITY i puna kompatibilnost sa FX-80 štampačem
OPCIJE :	Centronics (paralelni) interfejs 95$, RS-232 (serijski) 150$, IEEE-488 interfejs 150$, valjak za vođenje lista papira 60$, jednostruki magacin za smeštaj i opsluživanje papirom 499$, dvostruki 839$
CENA :	1395$ (osnovna konfiguracija)

Do sada smo, uglavnom, pisali o štampačima skromnijih karakteristika ali, što je važnije, pristupačnim po ceni širokom krugu naše, već „kompjuterizovane", nacije. Međutim, sve je veći broj onih kojima je računar ono što je nekada bila pisaća mašina — neophodno sredstvo za efikasniji i kvalitetniji rad. Ovde prvenstveno mislimo na novinare, književnike, prevodioce i sve one kojima je pisanje i obrada teksta svakodnevna preokupacija. Njima, i ne samo njima, potrebno je nešto više od jeftinog matričnog štampača na kojima hakeri razvijaju svoja programska remek-dela.



Slabije od foto-aparata, bolje od pisaće mašine: Dok u preciznosti crtanja ipak zaostaje za fotografijom, kvalitet otiska na novom „epsonu" može se meriti ne samo sa najboljom pisaćom mašinom nego i sa štamparskim slogom

Letter Quality printing

Our dream for the future is international. We are moving fast toward our goal of providing the very best and most advanced computer products to people the world over.

Let me invite you to find our more about us and see just why we believe that Epson means excellence in computer products.

PREGLED VISOKO KVALITETNIH MATRICNIH STAMPACA

PROIZVODJAC	MODEL	CENA ($)	PINOVI GLAVE	BRZINA S	BRZINA M	BRZINA LQ	MATRICA	GRAFIKA	BAFER	INTERFEJS	VALJAK	MAGAC PAPIRA	FORMAT	PROP STAM
ANADEX	WP-6000	2700	18	230	170	125	58x18	144x144	4.5K	CENT,RS232	100$	500$	SIROK	DA
BROTHER	DM40	1395	24	160	-	80	18x7/16x13	180x180	16K	CENT	DA	250$	SIROK	DA
CANON	PW-1156A	895	9	160	-	27	11x9/-/23x18	240 t/*	2K	CENT,(RS)	DA	NE	SIROK	DA
CENTRONICS	240	1495	24	160	-	80	7x18/-/30x16	180x180	132	CENT,(RS)	DA	280$	SIROK	DA
DATASOUTH	DS-220	1995	9	220	100	40	9x7/9x15/18x48	144x144	2K	CENT,RS232	DA	NE	SIROK	NE
DIABLO	32C@I	995	9	150	-	60	9x11/-/16x36	240 t/*	2K	CENT	NE	NE	SIROK	DA
EPSON	LQ-1500	1395	24	200	-	67	9x17/-/37x17	240 t/*	2K	(C,RS,IEE)	60$	499$	SIROK	DA
FLORIDA D.	OSP-130	3995	8	600	195	100	4x8/4x8/8x8	120x384	512	RS232,(C)	DA	DA	SIROK	DA
FUJITSU	DPL24	1995	24	288	160	80	9x24/13x24/36x24	180x180	4K	CENT,(RS)	DA	1400$	SIROK	DA
HEWLETT-P.	HP2934A	2895	12	200	67	40	9x12/36x24/-	90x90	2K	RS232	DA	NE	SIROK	NE
NEC	P3	1250	18	180	90	30	9x7/13x9/21x18	240x216	3.5K	CENT,RS232	175$	430$	SIROK	DA
PANASONIC	KX-P1093	899	9	160	80	31	9x9/9x9/17x18	216x240	2K	CENT,RS232	DA	NE	SIROK	DA
SANTEC	5700	3600	7	400	350	180	-/-/24x84	9e0x288	2K	CENT,RS232	200$	1100$	SIROK	NE
STAR MICR.	RADIX 15	995	9	200	-	50	9x9/-/18x18	240x144	16K	CENT,RS232	DA	NE	SIROK	NE
TEXAS INSTR.	TI-855	936	9	150	-	35	9x9/-/32x18	144x144	256	CENT,RS232	59$	499$	UZAN	DA
TOSHIBA	P1351	1895	24	160	-	78	16x11/-/24x24	180x180	4K	CENT,RS232	195$	1095$	SIROK	NE

BRZINA : S - STANDARDNI KARAKTERI ; M - "MEMO" KVALITET ; LQ - SLOVNI KVALITET
GRAFIKA : t/* - TACAKA PO INCU (2.54 cm)
INTERFEJS : CENT - CENTRONICS ; RS - RS232 ; IEE - IEEE-488
MAGAC : MAGACIN ZA OPSLUZIVANJE STAMPACA LISTOVIMA PAPIRA
PROP STAM : SPOSOBNOST STAMPANJA SA PROPORCIONALNOM SIRINOM KARAKTERA

jednom znaku, dok je kod drugih dovoljno jednom aktivirati pinove na glavi za štampanje.

Vratimo se našem LQ-1500. Pošto mu je osnova 60×37 cm a težina skoro 14 kg mora se planirati posebno mesto za njegovo smeštanje pri formiranju vaše računarske konfiguracije. Mora se priznati da dimenzije i težina ulivaju osećaj poverenja u ovaj štampač, što on u potpunosti zasluzuje.

Standardne karakteristike uključuju već pomenutu 24-to pinsku glavu za štampanje sa potpunom kontrolom svakog pina posebno, plastični vodič lista papira 15 formata 38 cm i punu kompatibilnost sa serijom FX Epsonovih štampača. Nagoveštena cena za osnovnu konfiguraciju LQ-1500 je 1395$ ali će, na žalost, morati da se izdvoji još dodatnih 95$ za centroniks interfejs. Opcijalno su na raspolaganju i RS-232 interfejs (150$) i paralelni IEEE-488, takođe po ceni od 150$. Što se tiče elemenata koji opslužuju štampač papirom, nudi se stvarno kvalitetan izbor: tzv. valjak za neperforiran papir za 60$, jednostruki magacin za smeštanje i distribuciju listova koji košta 499$, i na kraju, dvostruki magacin po ceni od 839$. Da napomenemo da se podrazumeva da štampač u osnovnoj verziji ima frikciono vođenje perforiranog papira.

Možemo izvesti i cenu po kojoj se može formirati zaista moćna kofiguracija: LQ-1500, plus jedan interfejs, plus jednostruki magacin za listove papira, plus „valjak", što sve ukupno iznosi oko 2100$. Možda ćete reći da ovo nije baš malo, ali svaka ozbiljnija upotreba ovog štampača će vrlo brzo opravdati investiranje u njegovu kupovinu.

Kontrola rada se vrši preko indikatora na gornjoj strani štampača. Opremljen je istim tasterima kao i ostali „epson" štampači: ON/OFF LINE, LINE FEED I FORM FEED, čije su funkcije u „Računarima" već bile razmatrane. Novina je taster SHEET LOAD koji otvara magacin listova papira i uvodi u nosač koji će držati papir za vreme štampanja. Normalno, cela prethodna operacija je potpuno automatizovana, tako da je izbegnuto stalno „punjenje" štampača papirom (šta ovo znači najbolje znaju oni koji su FX štampač koristili sa običnim tabakom).

Štampač LQ-1500 ima 3 osnovna seta znakova: obični ili standardni računarski (DRAFT), slovni kvalitet („LETTER QUALITY") i proporcionalno širok (PROPORTIONALLY SPACED) za najlepši ispis — prostor za „i" i „m" nije iste dužine. Svaki set ima 96 klasičnih, 96 „italik" (kosih) i 31 internacionalni znak. Štampač poznaje i srednji, tz. elitni, umanjen i proširen format

slova i znakove ispisuje naglašeno, dvostruko i naglašeno dvostruko u sva tri osnovna seta znakova. Izbor oblika slova se programskim putem (preko kontrolnih kodova kao kod FY serije) može vrlo jednostavno kombinovati. Posebno nas je impresionirao slovni kvalitet. Slova se teško mogu razlikovati od onih dobijenih na pisaćoj mašini ili klasičnom štamparskom tehnikom. Jednom rečju, ispis je vrlo blizak profesionalnoj klasi.

Što se brzine štampanja tiče, i ona je na zavidnom nivou. Tekst se ispisuje sa sjajnih 200 znakova u sekundi u običnom modu i sa oko 67 znakova u sekundi u „LETTER QUALITY" režimu. Smatramo da će ovi podaci ostaviti snažan utisak i na probirljivije korisnike računarske tehnike. Iako dosta brz, LQ-1500 je relativno tih štampač. I nešto što se posle ovoga ne bi očekivalo: on je projektovan kao veoma snažan uređaj, pa se može preporučiti i tamo gde su zahtevi za štampanjem dosta česti.

Matrica kojom se kreiraju znaci javlja se u 3 oblika. Za običan, standardni, izgled slova ona je tipa 9×17. Već ovo govori koliko je LQ-1500 bolji od svojih prethodnika. Da podsetimo: štampači iz FX serije su za istu namenu koristili matricu oblika 9×11. Za slovni kvalitet primenjuje se 15×17 tačaka. Na kraju, za slova proporcionalne širine matrica je formata 37×17 tačaka, što, ne bez razloga, pre možemo nazvati crtanjem a ne štampanjem. Inače, tekst se štampa dvosmerno u svim modovima, uključujući i slovni kvalitet, a svaka linija se generiše u jednom potezu.

I kod grafike imamo više mogućnosti za izbor. Jednostruka gustina podrazumeva 2,5 tačaka po horizontalnom milimetru, dok je taj broj u duploj ili četvorostrukoj gustini 5 odnosno 10 tačaka! Ovaj poslednji podatak sam za sebe dovoljno govori: pomoću štampača LQ-1500 može se veoma verno predstaviti čak i fotografija.

Da navedene karakteristike za štampač LQ-1500 ne bi ostale samo cifre koje mogu ali i ne moraju mnogo da kazuju, uporedili smo ih sa odgovarajućim osobinama štampača visoke klase. Rezultat upoređivanja je samo potvrdio naše opredeljenje za novi Epsonov proizvod. Brzina štampanja je ili ista ili bolja od brzine štampača čak i više cene od cene LQ-1500. Matrice koje ovaj štampač koriste pri kreiranju znakova su na nivou matrica i najskupljih matričnih štampača (pošto se radi o glavi za štampanje sa 24 pina, ovaj podatak nimalo nije iznenađujući). Grafička rezolucija je takođe u samom vrhu. Elementi koji se po slobodnom izboru mogu dodavati su i po asortimanu i po ceni na najvišem nivou. Sve u svemu, štampač LQ-1500 po karakteristikama u odnosu na cenu spada među najbolje, ako nije i najbolji u klasi matičnih štampača.

Na kraju, umesto zaključka: LQ-1500 preporučujemo svakome ko zahteva konstantnu upotrebu izdržljivog štampača za velike brzine standardnog ispisa, relativno visok slovni kvalitet, širok izbor grafike i mešano korišćenje teksta i grafike. LQ-1500 se jednostavno priključuje na bilo koji računar koji je danas u upotrebi, a sasvim solidan ulazni bafer kapaciteta omogućuje optimalno korišćenje i računara i štampača. Jednom rečju, on je brz, relativno tih i daje sasvim lepu kopiju. Šta više možemo poželeti? Jedino da bude jeftiniji i, normalno, da se može uvesti.

Srboljub Kuzmanović

Računari i programiranje

miladinova čarobna lampa

Prvo što razočaranom igraču pada napamet jesu komercijalni programi. Da biste ih koristili, potrebna je jaka motivacija: pisci će koristiti procesore reči, poslovođe i direktori tzv. „spreadsheet" programe i slično. Ovakvi programi znatno su skuplji nego igre i nisu nimalo laki za upotrebu. Najčešće zahtevaju i disk i štampač — a za mnoge naše vlasnike računara obe „stvarčice" su misaona imenica. Pa čak i kad ih imate, još uvek je problem kako to sve spojiti u harmoničnu celinu. (Jedno je sigurno: sa takvim problemima nikad neće biti dosadno u životu.)

Na žalost, broj programa koji se lako mogu koristiti kod nas prilično je ograničen. Strani komercijalni programi pomažu, ali ih ne treba precenjivati. Novopečeni vlasnik „spektruma" može steći utisak da je „sve već isprogramirano" i da „samo treba naći pravi program". U našim uslovima to, zasigurno, nije tačno. Nema naročitog smisla uzeti neki engleski program za tekući račun — naša bankarska pravila potpuno se razlikuju od njihovih. A da li je, možda, moguće promeniti nečiji tuđi program? Mogućno jeste, ali ugodno nije. Uostalom, šta da se radi kada je program isporučen na mašinskom jeziku? Hakeri bi se radovali ovakvom problemu, jer to shvataju kao neku vrstu intelektualnog sporta. Za prosečnog korisnika, intervencije u tuđem mašinskom jeziku samo su smetnja ka cilju, a nikako ne izvor razonode.

Stvar je pogoršana ponašanjem softverskih firmi koje sve čine da otežaju izmene u svojim programima. Naprotiv, njihova parola svodi se na: „Kupite naš program i rešite svoj problem pritiskom na dugme!" Cilj ovakve politike je da se korisnik dovede u krajnju zavisnost — praktično, kao da je kupio licencu za neki fabrički proizvod. Sve u svemu, postojeći programi rešavaju probleme koje su postavili njihovi autori, a (uslovnom) ne probleme kupaca! Pogledajte malo komercijalne programe za vaš računar — postoji li ijedan jedini čiji vam dizajn ne bi bar malo zasmetao? Naravno, nemoguće je efektivno koristiti računar bez ovakvih kupovnih programa. Mnogi vlasnici računara ostaju na ovom nivou, koristeći računare da urade poslove koji su ranije bili teško izvodljivi. Tako će jedan pisac znatno lakše i brže pisati na računaru nego da piše na klasičnoj pisaćoj mašini. Umesto dve ili tri revizije teksta, sada se može obaviti do desetak revizija! Krajnji rezultat je kvalitetniji tekst. Na sličan način računari pomažu i u projektovanju: firma „Westinghouse" koristi najbrži računar na svetu Cray-i da od miliona mogućnosti u projektu nuklearne centrale odabere optimalnu. Za kućni budžet dovoljno je da vam računar ponudi i samo dve-tri alternative. Računari korišćeni na ovaj način izrazito su korisne sprave.

Priča se ovim ne završava. Cela buka oko računara ne bi ni postajala da su oni samo zgodne pisaće mašine, ili jeftini surogati za sintisajzere. Ko se oduševljava novim modelima običnih pisaćih mašina? Računari nude i „nešto više", a mi ćemo pokušati da u ovom napisu odgonetnemo šta. Naša teza je: treba programirati!

Profesija zvana programer

I pre nego što su kućni i lični računari osvojili svet, postojali su veliki (i skupi) poslovni računari. Programiranjem su se na samom početku (50-tih godina) bavili isključivo matematičari i elektroinženjeri. Tada niko nije mogao da pretpostavi da će se pojaviti potreba da skoro i znaju da programiraju, kao što danas skoro svi znaju da voze kola. Ubrzo se otkrilo da računari mogu da izvedu bilo kakvu operaciju nad kodiranim informacijama, a ne samo da računaju Beselove integrale. Profesija programera postajala je sve brojnija, a sedamdesetih godina, dakle neposredno pre eksplozije kućnih računara, bilo je oko milion profesionalnih programera, među kojima su mnogi bili načisto zalutali u ovu profesiju. Programiralo se uglavnom u timovima, prosto zbog velikog obima posla. Tada je nastala sledeća podela procesa programiranja:

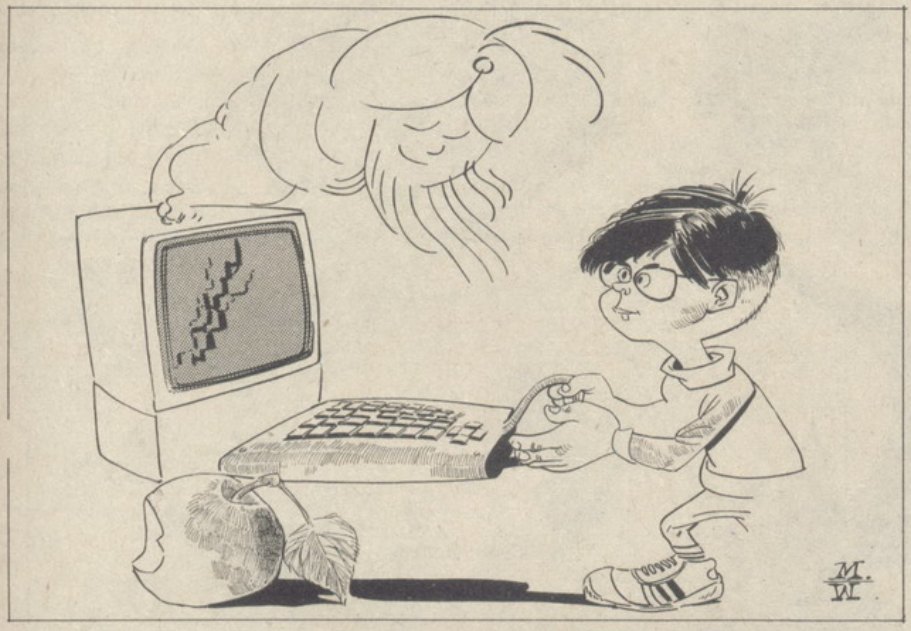
- Postavka problema
- Matematički model
- Algoritam
- Test-primer
- Pisanje programa (kodiranje)
- Izlazni rezultati
- Ocenjivanje izlaznih rezultata
- Dokumentacija

Specijalizacija je došla do vrhunca i za skoro svaki od ovih delova načinjene su posebne pod-profesije. Prva tri dela su daleko najprivlačnija u celom programiranju i njima su se zanimali tzv. sistem-analitičari, sa jedinim zadatkom da misle. Ostali su sve više postajali tehnička lica, a sa sve dosadnijim poslom.

Proces programiranja sastojao se iz ovih faza pre nego što su mikroračunari došli u milione domova širom sveta. Mikroračunari su promenili situaciju iz temelja, a faze procesa programiranja su se stapale. Programerski timovi postoje i danas, ali cena programa polako teži ceni jedne tvrdo-ukoričene knjige, proizvođači softvera sebe počinju da nazivaju izdavačima — po ugledu na izdavače knjiga, a programiranje ponovo postaje čisto individualna aktivnost. Uostalom, knjige uglavnom piše jedan čovek, a ne tim autora . . .

Prosečan „kućni programer" radi otprilike ovako:

Konačno ste se domogli svog prvog kućnog računara. Kakve li radosti! Posle mnogo dana štednje i odricanja i strepnje „da li će proći", to malo čudo tehnike stiglo je u vaš dom. Sa svih strana ste slušali izjave poput „računari su naša budućnost", „ako zaostanemo u razvoju mikroračunarske tehnike, postaćemo kolonija tehnološki najrazvijenijih sila" itd. Na pitanje „a šta ta sprava može da radi?" sležete ramenima i neubedljivo pokazujete na gomilu kaseta sa pokradenim igrama. Jer, naravno, prvo ćete se igrati — to su jedini programi stvarno jednostavni za upotrebu. A kada prvi zanos prođe, i kada doživite i više od 32.5% avanture u Hobitu — šta onda? Sve je više razočaranih kupaca „spektruma" i „comodora": igre su potrošna roba baš kao i pop-pesme. Kako, zaista, dalje? Zar je to ta budućnost o kojoj svi pričaju?



• Interaktivno kodiranje sa testiranjem, tj. napiše deo programa pa sa testira itd.
• Postavka problema

Faze su spojene u jednu, osim postavke problema, a dokumentacija se retko kad pravi. Kako je mogućno da se piše program a da postavka problema nije razjašnjena? Po svojoj prirodi, programiranje nas tera da budemo savršeni — što je izuzetno teško. Programi se sastoje od hiljada znakova, a svi moraju biti ispravni i osmišljeni. Tokom pisanja programa u interaktivnom režimu, sam proces kodiranja navodi nas da bolje upoznajemo problem o kojem je reč. Kada program jednom proradi, mi smo „uzgred" postali specijalisti i za objekat programiranja. To naravno ne znači da ćemo bolje trčati ako smo napisali program koji analizira rezultate nekog atletičara. Međutim, ako smo pisali program za simulaciju rada mozga — sasvim sigurno ćemo bolje znati kako mozak radi nego pre pisanja takvog programa. Drugim rečima, ako hoćete da postanete ekspert za neku oblast — počnite da o njoj pravite programe.

Programiranje i programirano učenje

U jednom vrlo konkretnom smislu, programiranje je najbolji metod programiranog učenja. Ovde nam preti mala terminološka zbrka: programirano učenje nije ni u kakvoj direktnoj vezi sa programiranjem računara. Programirana nastava bila je hit-ideja 60-tih godina, ali je ubrzo propala. Suština je bila da se učenik sam vraća na delove gradiva koje ne razume, i to u svom sopstvenom tempu. To je učenicima smrtno dosadno: „Vrati se na stranu 23. i ponovo proradi lekciju br. 3" — takve „zapovesti" doista su odbojne. Sličnost u procesu programiranja sa idejom programirane nastave je sledeća: tokom razvoja programa pravimo neke greške. Na neke nam ukazuje sam računar, a nekih nismo možda ni svesni. Kada otkrijemo grešku, tada se — isto kao u programiranom učenju — vraćamo na taj (prethodni) deo i ispravljamo ga. Kroz ispravku, mi učimo kako je to trebalo uraditi na samom početku, što je ekvivalentno ideji programiranog učenja da se vraćamo na „rupe" u svom ranijem znanju.

Kako to da nije dosadno i odbojno ispravljati greške u programu, a jeste dosadno i odbojno učiti stare lekcije? Odgovor je vrlo prost: mi smo pisali program, uložili smo u njega svoje vreme i znanje, pa nam je stalo da program proradi. I do sada su se računari koristili u nastavi, ali samo kao pomodna verzija već ranije poznatih „elektronskih učinioca", koje su radile po sledećem principu: nastavnik postavi pitanje a učenici pritiskanjem tastera ispred sebe tipuju na tačan odgovor; elektronska učionica (ili računar) vode evidenciju o svakom učeniku. Od korišćenja računara kao administrativnog sredstva trebalo bi preći na korišćenje računara kao sredstva za razvijanje zaključivanja, opažanja, mišljenja. Svaki program je esej i to u izvršnom obliku, tj. u obliku u kojem ga računar može izvršiti. Nije teško zamisliti da će u školi bliske budućnosti nastavnici učenicima zadavati — ne više domaće zadatke — već da pišu programe kod kuće. To bi (donekle) pojednostavnilo i ocenjivanje: prelaznu ocenu mogao bi dobiti samo učenik čiji program radi!

Matematičari posebnog kova

Opšte je poznato da su računari „genijalni idioti". Mogu se grubo porediti sa čovekom: po dolasku na svet, odnosno po uključenju mašine i čovek i računar se moraju obrazovati. Ljudi među životinjama, a računari među mašinama, jedine su nespecijalizovane vrste! Shodno tome, mogu se kasnije opredeliti i menjati „zanimanja" — svaki put kad računaru promenimo program. Mi ga „naučimo" da radi nešto drugo. Da bi sa „naučili", treba mu „objasniti" šta treba da „radi", što podrazumeva da naša htenja izrazimo tako da ih računar može razumeti. Konkretno, treba kodirati program na nekom programskom jeziku.

S druge strane, računari su napravljeni kao determinističke sprave: pod istim uslovima — isto rade. Programer treba da anticipira sve moguće događaje u programu, i da predloži tok akcije u svakom posebnom slučaju. Paradoksalno, programiranje „hladnih i ne-emocionalnih" računara zahteva kreativnost od programera. To je strogo kontrolisana vrsta fantazije — precizna i objektivna isto vreme. Programer mora da bude i vizionar i kritičar, a sve su to osobine koje krase vrhunske naučnike. Misaone operacije koje su potrebne u programiranju su:

• Apstrakcija — koristimo je svaki put kada izmislimo novu programsku varijablu.
• Razlikovanje slučajeva — zamišljamo svaku mogućnost koja se može dogoditi pod razumnim pretpostavkama: ova mentalna operacija efikasna je samo kada je broj slučajeva relativno mali.
• Zaključivanje po indukciji — potrebno je da razumemo petlje i situacije koje se programski ponavljaju.

Programiranje je posebno dobar uvod u matematički način mišljenja. Moguće je da „neki novi klinci" koji su se u predškolskom dobu igrali i upoznali sa računarima, iznađu da je matematika jedini lak predmet — jer ih podseća na računare. (Za sada, u SAD deca već počinju da se dele na one koji su odrasli sa računarima i — na one druge.) Evo na koji način je programiranje slično matematici:

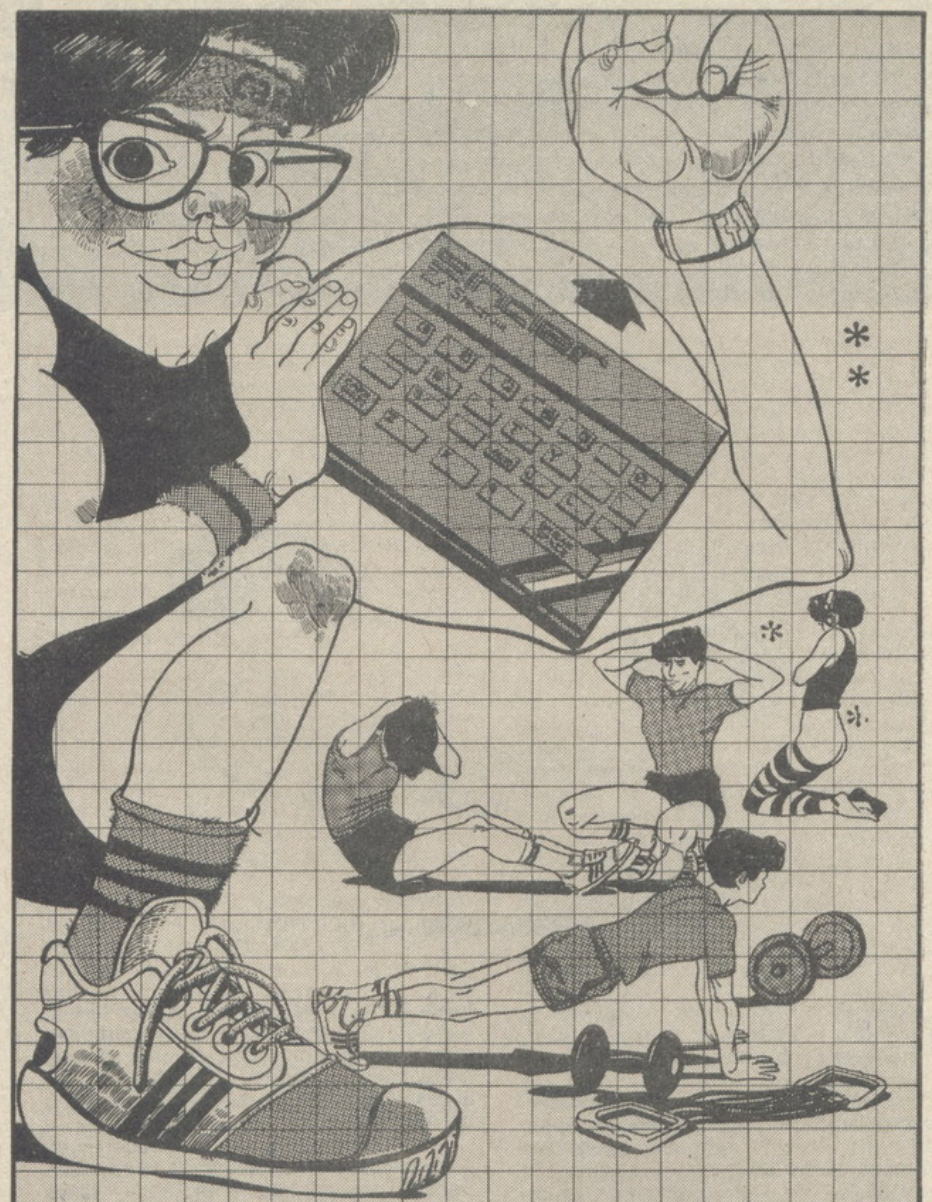
• Matematička tvrđenja uvek su opšta, odnoseći se na prebrojivo mnogo ili beskonačno mnogo slučajeva. Program je takođe definisan i radi za beskonačno mnogo slučajeva-kombinacija ulaznih veličina.
• Matematička tvrđenja veoma su precizna (što je neobično, jer se obično opštost postiže neodređenošću). Problem koji neki program rešava takođe je vrlo precizno formulisan, jer se računarima sve potanko mora „objasniti".
• Matematičko izlaganje je vrlo ubedljivo — stav, iskaz, dokaz itd. Slično se i programi mogu „dokazivati", tj. da se korisnik programa racionalnim rasuđivanjem ubedi u valjanost programa.

Nije, dakle, čudno što je po jednoj definiciji programiranje najteža grana primenjene matematike! Svaki programer je istovremeno, i matematičar, doduše od jedne posebne vrste. To nipošto ne znači da će vaš sin bolje rešavati kvadratne jednačine ako stalno programira svemirske igre ili avanture. Da bi se rešila kvadratna jednačina, treba prvo videti formulu za njeno rešavanje, tj. znanje činjenica i u matematici je presudno. No, zato bi vaš sin naučio kvadratnu jednačinu za sva vremena, ako bi napravio program koji radi za sve moguće vrednosti ulaznih podataka. (Napomena mladim programerima: ako hoćemo SVE moguće ulazne vrednosti, onda je ovo vrlo težak zadatak!)

Kraljevski put u nauku

Vaspitne vrednosti programiranja ovim se ne iscrpljuju. Pisanje programa, zapravo, predstavlja naučni čin, a programer je istovremeno i naučnik-istraživač. Rad sa računarom zahteva kristalno čisto i eksplicitno razmišljanje. U psihologiji je poznat termin pozitivni transfer. Učenje italijanskog je lakše ako ste prethodno učili francuski, ili bilo koji iz grupe romanskih jezika. Prosto rečeno, ponašanja iz jedne oblasti nesvesno se prenose u drugu. Veštine koje budete stekli kroz programiranje povoljno će se odraziti i u svim ostalim aktivnostima. Specijalno, vaše sposobnosti za rešavanje problema biće višestruko uvećane, a vaše mišljenje postaće zasnovano na činjenicama, na stvarnim podacima — a ne na željama i dobrim namerama. Jer programiranje pretpostavlja analitički pristup sopstvenim greškama, dakle, kritiku a ne kriticizam.

S druge strane, da bi napravio program, programer mora da ima viziju krajnjeg cilja, pa programiranje u sebi sadrži i postupak sinteze: ono je uvek svrsishodno. Programer zna tačno šta hoće u svakom trenutku, a ako hoće nemoguće — računar ga spušta na zemlju. Proces nastajanja programa je u suštini, naučni čin jer nauka nije ništa drugo nego naše stalno usaglašavanje sa postojećim, još neotkrivenim činjenicama. Naučnik svome problemu prilazi nizom hipoteza; programer konačnoj verziji programa prilazi kroz niz verzija programa. Baš kao što naučnik provodi eksperimente da bi potvrdio ili odbacio svoju trenutno aktuelnu hipotezu, tako i programer testira i eksperimentiše nad svojom trenutnom verzijom programa.



Ilustracija: MIŠA MARKOVIĆ

Granice ove analogije potpuno se brišu ako je program model nekog sistema kojeg ispitujemo. U tom slučaju, nova verzija programa bukvalno je i nova hipoteza o radu sistema. I tu je možda najveća vrednost računara, kojom su dopunili postojeći arsenal naučnih metoda: možemo napraviti netačnu hipotezu, proveriti je na računaru (tj. znatno brže nego bilo kojim klasičnim sredstvom eksperimentisanja), i odbaciti je ako računar tako odluči. Snaga računara je jednostavna: oni materijalizuju naše misli — i kao u savršenom ogledalu — vraćaju odraz, brzo i kristalno jasno. Na kraju procesa, i programer i naučnik dolaze do jedne zadovoljavajuće verzije naučnog objašnjenja neke pojave, odnosno programa na zadatu temu. Parafrazirajmo poznatu grčku repliku: „Programiranje je kraljevski put u nauku!" Računar je mašina koja je konačno oslobodila ljudsku kreativnost i približila je svakom vlasniku računara.

Sve to ima i svoju psihološku zasnovanost. Naučni (i programerski) pogled na svet polaze od realnih činjenica, što je preduslov da se taj isti svet može menjati. Naša kontrola nad spoljašnošću se povećava, i postajemo moćniji i sigurniji. Često se čuju primedbe da se današnja deca isuviše zaluđuju računarima. To je tačno ako se period igranja raznim igrama poput „pakmena" prekomerno produži, recimo više od godinu-dve dana. Verovatno se još nismo navikli na nova vremena — niko se ne buni ako deca mnogo čitaju, mada je po sredi ista stvar: kroz knjige čitalac vodi unutrašnji dijalog sa piscem koji je vremenski i prostorno udaljen. Ali, i preko računara (doduše, elektronski realizovano) programer vodi dijalog sa ljudima koji su projektovali taj računar ili program. Računari su, naravno, privlačniji jer pružaju sliku, boju i ton, delujući na naša čula mnogo konkretnije nego knjige, kojima je mašta jedini put do prihvatanja.

Ako smo se „zaljubili" u računare samo zahvaljujući igrama, kako li će tek prihvatiti računare oni koji će pomoću njih menjati svet?

Duško Savić

pisanje bez muke

Računari u primeni

Obrada teksta

Nijedan tehnički pronalazak posle Gutenberga nije toliko unapredio mučni zanat zvani pisanje kao što to čine (kućni) računari opremljeni specijalnim programima za obradu teksta. Iako su oni još uvek daleko od savršenstva „sedi-i-piši-ne-razmišljajući-kako-se-sprava-koristi", računare prihvataju ne samo sekretarice, novinari i prevodioci, kojima je presudna lakoća i brzina pisanja, nego čak i pisci, možda najveći tehnološki tradicionalisti, kojima je, iznad svega, bitan kvalitet teksta i koji su koliko juče bili spremni da prezru svako štivo koje nije napisano rukom i guščjim perom. Pri tome magija računara nije u lakoći pisanja. Njihova glavna snaga dolazi do izražaja tek kada, nakon što tekst „odleži", počne „muka sa rečima" — kad se odmerava svaka nijansa u značenju, a pojedine stranice prekucavaju nebrojeno puta. „Elektronske makaze" u računaru mogu u ovoj fazi da uštede i čitave mesece rada.

Uprkos tome što su se u toku svog postojanja računari neprekidno smanjivali po dimenzijama, polje njihove primene se širilo kao da su nastojali da opravdaju postojanje izreke da se u malim bočicama drži otrov. Tako se prvobitna upotreba u obradi numeričkih podataka proširila na obradu slike, teksta, glasa, laboratorijskih signala itd.

Jasno je šta predstavlja obrada numeričkih podataka i na kakve sve složene numeričke postupke možemo naići u rešavanju problema iz različitih konkretnih disciplina.

Ali, šta je obrada teksta? Šta se njome postiže? Koji su to tipični problemi koje treba rešiti i koje operacije treba definisati da se pomoću njih ti problemi rešavaju?

Smatrajući da mnogi čitaoci nisu imali neposrednog iskustva u ovoj primeni računara, ukratko ćemo opisati zahteve koje postavlja obrada teksta, tipične karakteristike programa za obradu teksta koji odgovaraju postavljenim zahtevima, i neke osnovne hardverske momente koje treba imati u vidu.

Dijalog sa mašinom

Jasno je da pisanje na mikroračunaru mora da teče sa lakoćom kao kada koristimo oruđe naših predaka — papir i olovku. Naravno, od računara se očekuje i mnogo više.

Za početak, olovku ćemo morati da zamenimo nešto skupljim uređajem, tastaturom. Umesto papira, koji smo do sada nemilice trošili, prelazimo na pisanje po ekranu. Opet skuplje, ali nam je, srećom, dovoljan samo jedan.

Dakle, tekst ćemo pomoću tastature unositi u računar. Svaki uneti simbol biće prikazan na ekranu. Da bismo mogli da određujemo gde želimo da upišemo pojedina slova ili reči, na ekranu se prikazuje kursor, obično kvadrat veličine polja za slova, koji označava gde će se prikazati sledeći simbol koji otkucamo na tastaturi. Kursor predstavlja vrh olovke.

Kako se upravlja ovakvom olovkom? Vrlo jednostavno. Na tastaturi postoje tasteri kojima pomeramo kursor po ekranu.

Dopalo bi se i Šekspiru: Od svih spisateljskih zanata, računari, možda, najviše unapređuju stvaranje književnih dela, jer se pisci, po prirodi stvari, od svih ljudi koji drže pero u rukama, najviše „muče sa rečima"

Pomeranje se može vršiti samo u četiri smera: gore—dole po vertikali, ili levo—desno po horizontali, sa koracima od najmanje jednog simbola. Na taj način nam je svaka tačka na ekranu dostupna.

Svakako da prostor jednog ekrana nije dovoljan za pisanje tekstova. Najčešće je potreban nešto širi i daleko duži.

Zamislite dugačku traku na kojoj ima mesta za upisivanje do recimo 240 simbola u jednom horizontalnom redu, i na koju može da stane jako mnogo redova jedan ispod drugog. Ako se ekran zamisli kao „prozor" koji možemo pomerati gore—dole i levo—desno po traci i posmatrati pravougaoni sektor koji je u prozoru, ili u taj sektor upisivati neki tekst, onda je to koncept kojim gotovo svi programi za obradu teksta rešavaju problem „tesnog" ekrana.

Dugačka traka, naravno, predstavlja unutrašnju memoriju u kojoj je tekst zapravo i smešten. Predstava o statičnom tekstu i ekranu koji se pomera, može se ravnopravno zameniti predstavom o statičnom ekranu i tekstu koji se pomera.

Ovaj dosta jednostavan koncept, na sasvim zadovoljavajući način, omogućava lako unošenje teksta na proizvoljnom mestu, jednostavnu „inspekciju" već unetog teksta, i ne ograničava veličinu teksta.

Poslovni sistem: Opremljen disk jedinicama i procesorom iz paketa „Torch", računar BBC B može da prihvati i profesionalni softver, uključujući i izvanredan program za obradu teksta „Wordstar"; cena ovakvog sistema je, međutim, veća od cene poslovnih sistema; ako se štampač zameni jeftinijim modelom, a pisac na računaru zadovolji i programom „Perfect Writer", koji je uključen u cenu paketa „Torch", troškovi nabavke neće preći 1500 funti (oko 450.000 dinara)

Elektronska gumica

Kako se pri pisanju i greši, verovatno da je gumica za brisanje izmišljena istovremeno sa papirom i olovkom (a možda se čak i bolje prodavala). Pored brisanja gumicom, vrlo često se i precrtava, dopisuju slova, koriguju promašeni padeži, ubacuju zaboravljene reči.

Računar je dužan da nam obezbedi mogućnosti za osnovne, „ručne", korekcije teksta. Od njega se, međutim, može zahtevati i znatno inteligentnija pomoć.

Bolji programi za obradu teksta obezbeđuju dva osnovna režima u kojima se izvodi ručno korigovanje teksta.

U prvom režimu, postojeći tekst ne „reaguje" na promene. Ako se upisuje nov tekst, on će preklapati stari koji se time uništava. Pri brisanju ostaju prazni prostori. Ovaj režim je naročito pogodan pri formiranju tabela.

U drugom režimu se postojeći tekst „adaptira" na naše korekcije. Ako se upisuju novi simboli, stari tekst desno od kursora se sa svakim unetim slovom automatski pomera udesno, oslobađajući, tako, mesto za novi tekst. Na taj način se ništa ne gubi. Pri ovom režimu umetanja, računar obavlja i sve ostale radnje koje su neophodne (sukcesivno prenošenje nastalog viška iz jednog reda u sledeći, eventualno potrebno pomeranje prikaza na ekranu i sl.). Ako se u ovom režimu briše, tekst desno od kursora se pomera i popunjava nastalu prazninu.

Pored ovoga, najčešće su obezbeđene i komande za brisanje reči (od razmaka do razmaka), redova, paragrafa, teksta od kursora do kraja i celog teksta iz memorije.

Ali, u računarskoj obradi teksta, pored ovog „ručnog" korigovanja, možemo zahtevati i automatsko.

Tako gotovo svaki program za obradu obezbeđuje nalaženje određenog dela teksta, nalaženje i brisanje, i nalaženje i umetanje. Računaru se, najpre, zada simbol, reč ili neki manji deo teksta koji treba pronaći. Po komandi, računar pretražuje tekst u memoriji, i kada pronađe ono što mu je zadato, pomeri ekran na taj sektor u tekstu, a kursor iza nađenog simbola ili reči. Sada se može intervenisati ručno. Na isti način se određuje i sledeća pozicija, ako se tražena reč ili simbol nalazi još negde u daljem tekstu.

Takođe se može zahtevati nalaženje, a zatim automatsko brisanje nekog dela teksta, ili nalaženje jednog dela i automatsko zamenjivanje nekim drugim zadatim sadržajem. Tako se na primer, može u celom tekstu reč „kompjuteri" zameniti rečju „računari" jednom komandom, ili neprijatna činjenica „nema para", više optimističkom „biće para, mada ih u ovom momentu još nema".

Kada tekst ukucamo, pregledamo i otklonimo greške, moramo ga smestiti na spoljnu memoriju, jer bi ga, inače, kada isključimo računar, nepovratno izgubili. Tekst nam je potreban ili radi dokumentacije, ili treba da ga nastavimo ako ga nismo završili, ili smo rešili da ga kasnije prepravljamo, ili ćemo ga štampati naknadno. Bilo koji da je razlog, pored upisa, trebaće nam i naredbe za učitavanje tekst datoteka sa diskete ili kasete. Ni jedan program za obradu teksta nije bez ovih mogućnosti.

Uređivanje teksta

Predpostavimo da smo tekst koji pišemo nekoliko puta dopunjavali i korigovali i da se na disketi (ili kaseti) nalazi poslednja verzija kojom smo zadovoljni. Sledeća etapa u radu na tekstu je njegova priprema za štampanje, odnosno formatizacija.

Formatizacija teksta je, zapravo, postupak njegovog oblikovanja prema našim zahtevima. Time se postiže da tekst u štampanju dobije oblik poslovnog pisma, stranice u knjizi ili završnog rada na kraju školovanja.

Formatizacija predstavlja „najubojitije oružje" programa za obradu teksta.

Koje su elementarne operacije dovoljne da se tekst može dovesti u bilo koju formu koja je potrebna u praktičnom radu? Evo najčešće korišćenih:

Definisanje leve i desne margine

Zadavanjem leve i desne margine postiže se da svaki odštampani red teksta leži unutar postavljene leve i desne granice, ali ne i obavezno na njima. Leva margina se najčešće bez posebne komande pri unošenju teksta automatski poravnava na zadatoj poziciji, a za desnu treba to posebno zahtevati.

Poravnavanje desne margine

To je postupak u kome se automatskim prebacivanjem reči iz nižih u više redove (grubo doterivanje) i dodavanjem potrebnog broja razmaka između reči u jednom redu (fino doterivanje), poslednji simboli svih redova teksta poravnavaju na istu vertikalnu kolonu.

Jeftino i kuso: Oko popularnog „spektruma" može se izgraditi najjeftiniji sistem za pisanje na računaru, ali je i ta cena, s obzirom na ograničenja, prevelika — izuzetno loša tastatura i mikrodrajvi umesto disk jedinice

Ova operacija značajno doprinosi lepšem izgledu teksta i može se koristiti i na samo pojedinim delovima teksta, dok se ostali delovi, ako je to potrebno, mogu ostaviti u onoj formi u kojoj su i uneti.

Centriranje

Pojedine redove teksta treba centrirati na papiru. Time se korisnik oslobađa mučnog prebrojavanja slova kada se kucaju naslovi i drugi delovi teksta kojima je mesto na sredini reda.

Uvlačenje reda (ili redova)

Za nove odeljke u tekstu, ili početke paragrafa, koristi se uvlačenje prvog reda za određen broj simbola. Ova pogodnost doprinosi čitljivosti teksta i mora se nalaziti u „repertoaru" format operacija.

Prored

Prored je rastojanje između dva uzastopna reda teksta. Od njega zavisi koliko će redova biti odštampano na jednoj stranici. Najčešće su na raspolaganju jednostruki, dvostruki i trostruki prored (kao kod pisaćih mašina), ali se često prored može podešavati i sa finijim koracima.

Dužina teksta

Dužina teksta na jednoj štampanoj stranici meri se brojem redova. Može se zadati po želji, ali ne može biti veća od maksimalnog broja redova, koji zavisi od fizičke dužine korišćenog papira i proreda.

Straničenje i označavanje

Kako računar sam deli tekst na stranice prema formatu koji smo mu zadali, prirodno je da on vrši i obeležavanje stranica. Broj stranica možemo postaviti na vrh ili pri dnu. Ali, to nije sve. Većina programa za obradu teksta daje i dodatnu mogućnost da se, pored broja, stranica obeleži i nekim tekstom. To je mogao biti naziv teksta, mesto i datum kada je pisan, ime autora ili bilo šta drugo. Na taj način svaka stranica ima oznaku pomoću koje znamo kom tekstu pripada.

Ipak samo kompromis: lako u ovom času predstavlja možda najracionalniji izbor, računar „komodor 64" neće previše usrećiti pisce na kompjuteru — njegova disk jedinica je veoma spora, a kvalitet otiska štampača izuzetno loš

Ova korisna mogućnost upotrebljava se veoma jednostavno. Tekst kojim želimo da označimo svaku stranicu i položaj tog teksta (na vrhu stranice, na dnu stranice, ili jedna oznaka na vrhu a druga na dnu stranice) navodi se samo jednom, na samom početku.

Bezuslovna i uslovna sledeća stranica

Jasno je da je neophodno imati na raspolaganju mogućnost da se zahteva prelazak na početak sledeće stranice kada to želimo. Ali šta je uslovna sledeća stranica? Veoma jednostavno, to je stranica na čiji se početak prelazi ako na tekućoj stranici nije ostalo onoliko slobodnih redova koliko smo naveli. Dakle, ako nam u tekstu treba, recimo, 10 uzastopnih redova koje zaprema tabelu koju smo ukucali i koja ne sme da se deli na dve stranice, onda se pre početka tabele traži sledeća stranica sa uslovom 10. Ako tabela ne može da stane, računar sam započinje novu stranicu štampanjem tabele.

Postoji još niz dodatnih mogućnosti formatizacije koje se mogu sresti. To su podvlačenje, vraćanje glave štampača unazad, slanje komandnih sekvenci štampaču, dvokolonsko štampanje itd.

Formatizovanje se vrši ubacivanjem odgovarajućih naredbi u tekst. Na taj način je tekst „prošaran" dodatnim simbolima naredbi, ali se pri štampanju one ne pojavljuju. Format naredbe se kod nekih programa za obradu teksta moraju upisivati u posebnom, ubačenom redu, ali se često mogu i mešati sa tekstom u istom redu.

Štampanje teksta

Za štampanje teksta nije potrebno imati na raspolaganju veliki broj naredbi pošto je glavni deo posla već obavljen formatiranjem. Ali, kako štampanje dugo traje i troši papir, treba imati mogućnost pregleda formatiranog teksta pre nego što ga definitivno odštampamo. To se postiže naredbom za izdavanje formatiranog teksta na ekranu.

Ova naredba prikazuje na ekranu tekst u potpuno istom obliku u kome će se pojaviti na papiru. Vrlo je podesna kao poslednja kontrola pri kojoj vidimo efekte svih naredbi formatizacije. Ako smo zadovoljni, prelazimo na štampanje.

Veoma je korisno ako program za obradu teksta omogućava da biramo hoćemo li neprekidno štampanje ili ćemo, pritiskom na dirku tastature, posle svake odštampane stranice zahtevati sledeću. Ako pri tome još možemo interaktivno u toku štampe usmeravati tekst na ekran ili nazad na štampač, onda imamo mogućnost štampanja samo pojedinih stranica. To je potrebno kada nam je jedna stranica, recimo, zaprljana, a neka druga nema sve otkucane simbole jer se traka na štampaču bila privremeno pomerila. Tada štampanjem strana-po-strana i prelaženjem sa štampača na ekran i obrnuto, ponovo odštampamo samo te dve stranice.

Poželjno je na raspolaganju imati i komandu kojom određujemo broj primeraka teksta koje treba odštampati. Tako, dok vi negde honorarno radite da biste isplatili kupljeni kućni računar, tekst će biti proizveden u željenom broju primeraka.

Romani u nastavcima

Ako posebno ispišemo sve podnaslove od početka ovog članka, dobićemo tipičan redosled postupaka u obradi teksta, i grub opis osnovnih zahteva na koje programi za obradu teksta moraju da odgovore.

Međutim, postoji još čitav niz veoma korisnih i zanimljivih mogućnosti i operacija koje možemo očekivati od boljih programa. Pre svega, važna je mogućnost neposredne komunikacije sa onim delom operativnog sistema računara koji je zadužen za opsluživanje disk jedinice. Ako nam program stavlja na raspolaganje ove naredbe, onda smo u stanju da pripremimo novu disketu za rad, da pregledamo spisak imena svih programa i datoteka koji se nalaze na disketi, kao i da menjamo pojedina imena, da brišemo tekst datoteke ili neki drugi sadržaj sa diskete, da kopiramo diskete ili njihove delove itd. Svi ovi uzgredni poslovi „održavanja kuće" neobično su potrebni.

Često je kapacitet unutrašnje memorije nedovoljan za tekst koji pišemo. Radi toga treba imati na raspolaganju mehanizam povezivanja i štampanja više tekst datoteka kao jedne celine. Time se tekst koji je prethodno upisan na disketu u „porcijama" koje odgovaraju kapacitetu unutrašnje memorije, povezuje i može neprekidno štampati. Ovaj koncept povezanih tekst datoteka, koji se realizuje na različite konkretne načine, daje korisniku celu disketu za smeštaj jednog teksta, recimo knjige.

Jedna mogućnost, koju smatramo važnom, može se nazvati prohodnost ka drugim softverskim sredinama. To, praktično, znači da se tekst koji smo uneli, može koristiti i iz programa za baze podataka, na primer, ili jednostavno iz bejzika. To u najvećoj meri zavisi od načina kodiranja pri zapisu na spoljnu memoriju. Ako je to ASCII, standardni kod, i tekst datoteka organizovana sekvencijalno, onda je stvar jednostavna. Takvim se tekstovima lako pristupa, recimo iz bejzika, i nad njima vrše operacije koje ne možemo izvesti u okviru programa za obradu teksta. To bi moglo biti azbučno sortiranje, kodiranje i dekodiranje teksta da bi se onemogućili neželjeni „čitaoci", i slično.

Na kraju pomenimo još samo dve, veoma moćne nadgradnje koje se sreću samo kod najozbiljnijih programa za obradu teksta. To su tzv. makro naredbe i čitavi, doduše jednostavniji, specijalizovaniji programski jezici kao delovi mogućnosti programa za obradu teksta. Time se u, pravom smislu, reči postiže otvorenost prema korisniku, ali detaljniji opis odlažemo za neku drugu priliku.

Mašinska oprema

Hardver (ili, drugačije rečeno, mašinska oprema) neophodan za poslove obrade teksta sastoji se od računara sa tastaturom, monitora ili kućnog televizora, disk jedinice (mogao bi da posluži i kasetni magnetofon, ali bi to trebalo izbeći ako se može) i, na kraju, štampača.

Ne vezujući se za konkretne mikroračunare ili štampače, navešćemo neke osnovne opšte zahteve i probleme koje treba imati u vidu pri procenjivanju hardverskih konfiguracija.

Tastatura mora biti profesionalna, sa standardnim rasporedom slova i kvalitetnim dirkama. Ako taj uslov nije ispunjen, unošenje tekstova se pretvara u pravo mučenje. Oni koji su duže radili na „spektrumu" to vrlo dobro znaju.

Računar mora imati dovoljno unutrašnje memorije da može da prihvati program za obradu teksta i da na raspolaganju ostane dovoljno slobodnog prostora za tekst. Kapacitet od 2—3 kucane strane sigurno nije dovoljan. Treba reći da jedna standardna kucana strana sa proredom ima $60 \times 30 = 1800$ simbola, što zauzima 1800 bajtova. Znači da 10 kucanih strana zahteva 18 K slobodne unutrašnje memorije.

Za računar koji ste kupili mora postojati i program za obradu teksta koji zadovoljava vaše potrebe. To je, naravno, stvar lične procene, a u ovom broju časopisa se mogu

naći pregledi nekih od najpopularnijih programa.

Za računar je vezan i tipičan problem prikazivanja nedovoljnog broja simbola u jednom redu na ekranu. Ako se uzme da je 80 simbola u redu dovoljan broj, onda je tipičnih 40 ili čak 32 simbola sigurno nedovoljno. Ali, ovo ne znači da je, recimo „komodor 64" sa svojih 40 slova u redu neupotrebljiv za obradu teksta. Ovaj problem se svodi na apsolutno zanemarljiv nivo pogodnim rešenjima u okviru programa, kao što je to učinjeno u „Easy Script-u", verovatno najboljem programu za obradu teksta na računaru „komodor".

Štampač je završna i možda najdelikatnija karika u lancu. U najširoj su upotrebi dve vrste štampača zasnovani na dva različita principa rada — matrični štampači i štampači sa lepezom.

Matrični štampači sa svojom matricom nezavisno pokretnih iglica koje formiraju slovo pomoću otisnutih tačaka rade na „inteligentnijem" principu od štampača sa lepezom. Zbog toga su fleksibilniji — mogu samo na osnovu programske podrške, bez zamene mehaničkih delova, da štampaju razne veličine slova, razne tipove slova ili, čak, da omoguće korisniku da programira sopstveni skup simbola. Oni su, istovremeno, veoma brzi i ne naročito bučni. Mana im je to što se vidi tačkasta struktura slova na odštampanom tekstu. Ali, i ova mana se prevazilazi u novoj generaciji matričnih štampača sa velikom gustinom matrice za štampanje i visokim kvalitetom slova. Ovi NLQ (near letter quality) štampači ostaju i dalje prihvatljivi po ceni koja se kreće od 300 engleskih funti pa na više.

Štampači sa lepezom daju veoma visok kvalitet štampe, ali su spori, skupi i bučni. Oni nemaju fleksibilnost matričnih štampača, već im se samo može ručno menjati lepeza sa ugraviranim simbolima.

Pomenimo i jedan specifičan problem vezan za obradu teksta na našem jeziku — latinična slova č, ć, ž i š koja računari i štampači iz uvoza ne poznaju. Ovaj rproblem koji se proteže od tastature, preko skupa karaktera u računaru, do karakteristika štampača, može se lako rešiti pod uslovom da matrični štampač ima programabilne karaktere i da program za obradu teksta obezbeđuje njihovu upotrebu. Ali, ako koristimo štampač sa lepezom ili je program za obradu teksta „zatvoren", onda se ovi problemi rešavaju teže. Za neke tipove štampača sa lepezom čak se mora naručivati posebno graviranje traženih simbola. Da ne govorimo o ćiriličnoj azbuci.

Cilj ovog teksta nije bio da predlaže određeni program za obradu teksta, niti neki mikroračunarski sistem kao naročito pogodan. Opisane tipične karakteristike koje se mogu sresti kod ovih programa, kao i pomenuti hardverski detalji bitni u obradi teksta, mogu poslužiti za samostalno procenjivanje i odabiranje pri nabavci.

Veljko Spasić

Rečnik žargona

Ovaj sasvim kratak rečnik termina vezanih za oblast obrade teksta na računaru sadrži i kratka objašnjenja osnovnih pojmova. Njegov je cilj da olakša snalaženje u bujici engleskih izraza koji vladaju računarskim žargonom. Radi toga je uređen po engleskim rečima.

ALIGNED — PORAVNAT

Ivica teksta je ravna, tj. dodati su razmaci između reči u radu tako da se prvo i/ili poslednje slovo u svakom redu nalazi na istoj poziciji (koloni).

APPEND — DODATI

Učitati novi tekst iza kraja teksta koji je trenutno u memoriji. Na taj način se tekstovi nastavljaju.

COMMAND MODE — KOMANDNI REŽIM

Režim rada programa za obradu teksta u kome korisnik izdaje komande koje se izvršavaju.

CONDITIONAL FORCED PAGE — USLOVNA SLEDEĆA STRANICA

Ako pri izdanju teksta, na tekućoj stranici nije preostao onaj broj slobodnih redova koji je tražen u format naredbi za uslovnu stranicu, prelazi se na sledeću stranicu.

CONTINUOUS PRINT — NEPREKIDNO ŠTAMPANJE

Tekst se štampa stranica za stranicom, bez pauze.

CURSOR — KURSOR

Specijalan simbol (obično kvadrat) koji svojim položajem na ekranu ukazuje gde će se prikazati sledeći znak otkucan na tastaturi. Korisnik može menjati položaj kursora.

CURSOR CONTROL KEYS — DIRKE ZA KONTROLU KURSORA

Dirke kojima se kursor pomera po ekranu.

DAISY WHEEL PRINTER — ŠTAMPAČ SA LEPEZOM

Štampač koji koristi disk po čijem obodu su izgravirani simboli. Disk brzo rotira sve dok se traženi znak ne dovede u položaj ispred malog čekića koji udarom otiskuje slovo na papir.

DECIMAL TABS — DECIMALNA TABULACIJA

Način tabuliranja kojim se pri unošenju poravnavaju decimalne tačke tako da leže jedna ispod druge (u istoj koloni).

DEFAULT VALUES — VREDNOSTI KOJE NEDOSTAJU

Vrednosti parametara koje program za obradu teksta sam određuje ako korisnik nije drugačije odredio.

DELETE — PREBRISATI

Uklanjanje dela teksta bez ostavljanja praznog prostora. Ostatak teksta se pomera i popunjava prazninu.

DISK MODE — DISK REŽIM

Režim u kome se u okviru programa za obradu teksta izvršavaju operacije sa disketom.

DOT MATRIX PRINTER — MATRIČNI ŠTAMPAČ

Štampač koji koristi grupu pokretnih iglica da formira karaktere koji se štampaju.

EDIT MODE — REŽIM UREĐIVANJA

Režim u kome se neposredno vrše operacije sa tekstom. Ovaj režim je osnovni i najviše se koristi.

ERASE — OBRISATI

Uklanjanje dela teksta uz ostavljanje praznog prostora koji je tekst zapremao.

ESCAPE SEQUENCE — ESCAPE (ISKEJP) SEKVENCA

Niz simbola koji započinje simbolom ESCAPE (27 ASCII) čijim se slanjem štampaču aktiviraju njegove dodatne mogućnosti.

FILL FILE — DATOTEKA ZA POPUNJAVANJE

Tekst-datoteka čijim se sadržajem popunjavaju blokovi u tekstu koje je korisnik odredio.

FOOTING — DONJA OZNAKA STRANICA

Red uvek istog teksta koji se pojavljuje na dnu svake stranice. Može uključivati broj stranice.

FORCED PAGE — BEZUSLOVNA SLEDEĆA STRANICA

Kada program za obradu teksta naiđe na format komandu za bezuslovnu sledeću stranicu, onda se izdavanje teksta nastavlja na početku sledeće stranice.

FORMAT — FORMAT

Način na koji je tekst uređen kada je odštampan.

FREE FORMAT INPUT — UNOŠENJE U SLOBODNOM FORMATU

Unošenje teksta bez označavanja gde su krajevi redova.

HEADING — GORNJA OZNAKA STRANICA

Red uvek istog teksta koji se štampa na vrhu svake stranice. Može sadržati i broj stranice.

INSERT — UMETANJE

Postupak kojim se između dva susedna slova ili dva susedna reda unosi nov tekst.

JUSTIFICATION — IZRAVNAVANJE

Postupak kojim se leva i/ili desna ivica teksta dovodi u poravnati položaj. Program za obradu teksta sam vrši ovu operaciju.

KEYBOARD — TASTATURA

Ulazni uređaj računara preko koga se tekst unosi u unutrašnju memoriju.

LINKED FILES — POVEZANE TEKST DATOTEKE

Više tekst datoteka može se povezati u niz i za neke operacije tretirati kao jedan dokumenat. Time se prevazilazi ograničenje koje nameće kapacitet unutrašnje memorije.

LINKING SPACE — POVEZUJUĆI RAZMAK

Posebna vrsta razmaka između reči koji naznačuje programu za obradu teksta da se dve reči povezane ovim razmakom moraju štampati u istom redu.

PITCH — BROJ ZNAKOVA PO INČU

Pre štampanja teksta, može se kod nekih štampača, zadati broj karaktera koje treba smestiti na jedan inč. Time se reguliše horizontalna gustina odštampanog teksta.

RAGGED — NEPORAVNAT

Tekst kod koga je bar jedna ivica teksta ostala neuređena, tj. nije izvršeno poravnavanje margine.

SCROLL — POMERANJE (KLIZANJE)

Ako se tekst zamisli kao dugačka traka sastavljena od paralelnih redova poređanih jedan ispod drugog, onda se prikaz teksta na ekranu može zamišljati kao pomeranje (klizanje) ekrana po tekstu. Ovo pomeranje može biti kako vertikalno tako i horizontalno. na taj način se može pregledati čitav tekst.

SOFT HYPHEN — USLOVNI PRENOSNIK

Simbol koji se umeće između dva susedna slova jedne reči i naznačuje programu za obradu teksta da, ako se ta reč ne može cela odštampati na kraju reda, onda se njen deo desno od uslovnog prenosnika može preneti u sledeći red. Ova opcija se koristi za deljenje reči na slogove na kraju reda.

STATUS LINE — STATUSNI RED

Red na ekranu (obično najviši) rezervisan za prikaz važnijih informacija o stanju u kome je trenutno program za obradu teksta.

VDU — EKRAN

Skraćenica od „Visual Display Unit" — jedinica za vizuelni prikaz. Najčešće ekran televizora ili monitor.

WORDPROCESSING — OBRADA TEKSTA

Procedura u kojoj se tekst unet u računar obrađuje pomoću posebnog programa za obradu teksta.

Veljko Spasić

pirati trče počasni krug

Računari iz mog ugla

Pirati piratаju, hakeri hakiraju, a šta vi radite?

Pirati su nekad bili strah i trepet mora: banda ružnih prljavih i zlih što su otimali, silovali i ubijali — koncentrat kriminala na jednom brodu. Nije iznenađujuće da softverske firme sve one koji neovlašćeno kopiraju programe i time ih ometaju u zgrtanju profita pokušavaju da prikažu kao kriminalce. Čudno je kada neki domaći ljubitelji pravde počnu da se bore protiv sirotih malih pirata.

Svoju netrpeljivost prema piratima mogu da potkrepe varijacijama jednog istog argumenta: kopirajući programe, pirati rade protiv sebe, jer više neće biti kvalitetnih programa, pošto će softverske firme propasti. Kratko rečeno, pirati rade sve u korist svoje štete.

Baš lepo što ima ko da vodi računa o sudbini kapitalističke privrede koja nas je toliko zadužila! Pruža nam se prilika za vrhunski mazohistički ugođaj: kad već plaćamo najskuplji hardver na svetu, zašto nas ne bi opljačkali i na softveru?

Društvo za ukidanje pirata prima članstvo sa svih strana i dobro im dođe svako ko može nešto da doprinese. Oni koji preterano brinu o usmeravanju omladine na pravi put postaće istaknuti članovi tog društva. Pošto nemaju dovoljno čvrste argumente, preporučujemo moralisanje u fazonu: „Hakeri i pirati? Peting sa kućnim ljubimcem! To je, ako ne gnusna perverzija, a ono bar opaka boleština koju treba ukinuti energičnom akcijom milicije."

A pirati piratаju, i ne slute šta im se sprema.

Već je na pomolu kampanja da se suzbiju male štetočine koje podrivaju veliku privredu. Donet je zakon o zabrani presnimavanja audio i video kaseta, a sada bi neki da iskoriste tu gužvu i poture isto i za softver. Pozvaće se u pomoć i statistika koja će otkriti da već sada ima dva pirata po glavi stanovnika. Stvar se postavlja tako kao da su mali pirati otimači tuđeg dobra. Mali pirati — veliki lopovi.

Ovo nije čak ni zamena teze, već je u pitanju jedna od nižih vrsta logičkih podvala. Kod presnimavanja programa ne radi se o računskoj radnji ODUZIMANJA, kako bi to neki hteli da predstave, nego MNOŽENJA.

Mali pirati i hakeri lepo se zabavljaju dok razbijaju zaštitu i razmnožavaju programe. Skidanje mraka sa programa pruža im vrhunsko zadovoljstvo. To je izazov; večito nadmudrivanje između proizvođača i razbijača. Srećom, sigurno kontraceptivno sredstvo protiv razmnožavanja programa još nije pronađeno.

Oni koji brinu da u društvu sve funkcioniše kako treba znaju da je nekontrolisano zadovoljstvo opasno. Najbolje stvari su džabe, zato oni moraju pažljivo da motre. Oni se pojavljuju tamo gde ih najmanje očekujete i tamo gde im uopšte nije mesto. Zato pazite šta radite u spavaćoj sobi, kupatilu i dok se družite sa svojim kućnim ljubimcem.

Magičnu formulu sreće, zadovoljstvo +korist, ipak nisu pronašli ni mali pirati ni hakeri. Taj obrazac odavno je poznat u kapitalizmu, gde se interpretira ovako: korist naša — radost vaša.

Nije neophodno dobro poznavati tajanstvene zakone kotrljanja novca da bi se shvatilo ko proždire najveći deo kolača. Dok gospodare tržištem, zbog velike konkurencije, mali pirati se sigurno ne mogu obogatiti.

Nekada su najveću korist iz prohibicije alkohola izvlačili gangsteri. Sada su veliki

Ilustracija: Miša Marković

pirati najveći zagovornici opšte prohibicije piratije. Oni to rade vatrenije čak i od samih proizvođača softvera. U opštoj psihozi grabeža da se na kompjuterskoj groznici što više profitira, veliki pirati su najglasniji u povicima: „Drž'te lopova!". Jedino je čudno što se još uvek nisu pojavili profesionalni kerberi koji će za šaku dolara krenuti u lov na ucenjene glave sirotih malih pirata. Za Gargamela uvek ima posla.

Ti preprodavci na veliko žele da drže tržište u svojim rukama. Glavno zanimanje im je otimanje para deci pod parolom popularisanja računara među omladinom. Ne propuštaju priliku da se ubace u sve mas-medije. U sumnjivim novinama dele sumnjive savete, gostuju na radiju i slikaju se za televiziju da bi narod mogao da prepozna svoje prosvetitelje. Oni će vam uz ono što vam treba 'ladno uvaliti i žestoko naplatiti i ono što vam ne treba. Njima je potrebna velika publika, koju će da zavlače. Svoj zgrtački nagon pokušavaju da prikažu kao društveni interes. Oni su kontrarevolucionari računarske revolucije.

Trenutno velikim piratima najviše smetaju mali oglasi u kojima se programi nude u bescenje. Međutim, mali pirati ne daju se baš tako lako ukinuti. U njima teče hajdučka krv. Po najboljim tradicijama našeg naroda, po ugledu na hajduk Stanka, malog Radojicu i dijete Grujicu, pirati su kadri stići i uteći, i na strašnome mestu postojati. Oni otimaju od bogatih i vrše distribuciju softverskog blaga među sirotinjom rajom.

Korist od piratije nije samo u preraspodeli programa, nego i u osposobljavanju prave vojske mladih ljudi koja će sa tim softverom da se nosi. Strateški interes radnog naroda je pripremiti generaciju koja raste za 21. vek, gde će se kročiti smelo, a ne biti šlogiran i smrznut od straha pred trajvanima, kompjuterima i ostalim čudesima koja ih tamo očekuju.

* * *

P.S.

Od sada će kompjuterske novine umesto malih oglasa objavljivati poternice.

Kad malo bolje razmislim, to presnimavanje softver programa stvarno nema smisla. Ne dajte da vas zavedu ovakvi tekstovi. Prekinite već jednom sa piratanjem softver programa, kad vam lepo kažem.

Jelena Rupnik

biblioteka programa

Spektrum

Pretraživanje (search)

Dok pišemo neki program za računar, obično smo unapred pripremljeni na to da nam prva verzija neće proraditi. Tek posle startovanja programa, kada postane očigledno da se ne dešava ono što smo hteli, primetićemo da smo, na primer, u nekom potprogramu koristili promenljivu koja je već zauzeta u glavnom programu, ili da smo svuda pogrešno pisali ***GO TO 1000,*** *umesto* ***GO TO 2000,*** *itd.*

„Spektrumov" editor ne pruža bog zna kakve mogućnosti u fazi prepravljanja programa. Nedostaju naredbe za prenumeraciju, izbacivanje pojedinih blokova programskih linija, ili pretraživanje teksta u cilju zamene nekih delova. Na korisniku ostaje ili da se pomiri sa ograničenjima editora, ili da sistem dopuni kratkim mašinskim rutinama, koje će obavljati željene operacije. Program SEARCH, koji ovde objavljujemo, predstavlja jedno takvo proširenje, pomoću koga se mogu pronalaziti proizvoljni nizovi znakova u bejzik listingu i po želji izbacivati ili zamenjivati. Pretraživanje se vrši u unapred zadatom bloku programskih linija, a svaka linija se posle izmene podvrgava kompletnoj proveri sintakse.

Rutina je dugačka 440 bajtova, a pisana je tako da se može startovati sa bilo koje adrese. Posao oko generisanja mašinskog koda poveren je bejziku čiju listu objavljujemo. Dvadesetak sekundi posle njegovog starotvanja sa RUN, pojaviće se poruka „Start tape, then press any key", a mašinski program će biti poslat na kasetu. Odmah zatim će se obaviti i verifikacija snimka — samo treba premotati kasetu i ponovo je startovati.

Kada budemo želeli da koristimo ovaj program, uzećemo ga sa kasete i uneti negde u memoriju računara, najbolje iza RAMTOP-A. Iskusnim programerima tu više ne treba ništa objašnjavati. Međutim, pošto pretpostavljamo da ima više početnika koji se još ne snalaze sa terminima kao što su „adresa", „memorija" itd., predlažemo sledeću proceduru za unošenje mašinskog programa sa trake:

CLEAR 64927: LOAD "" CODE 64928 /48K/

odnosno:

CLEAR 32159: LOAD "" CODE 32160 /16K/

Ovim postupkom će se 440 bajtova programa SEARCH smestiti u sam vrh memorije, iznad RAMTOP-a, a ispod grafičkih znakova.

Rutina SEARCH zahteva na ulazu četiri podatka:

1. Početnu liniju „s" bloka koji se pretražuje.

2. Konačnu liniju „e" bloka koji se pretražuje.

3. Niz znakova „a$" koji se traže.

4. Niz znakova „b$" kojima se zamenjuje prethodni niz.

Prosleđivanje ovih parametara vrši se metodom DEF FN funkcija, o čemu smo već pisali u „Računarima 3". U ovom konkretnom slučaju, funkcija će imati oblik:

DEF FN f(s,e,a$,b$)=USR 64928

Najjednostavnije je ubaciti ovu definiciju na sam početak programa, recimo u liniju 10, mada je, naravno, sasvim svejedno gde će ona biti. Negde u programu se, ipak, mora naći. Ne treba posebno ni naglašavati da ime funkcije ***f,*** *kao i imena parametara* ***s, e, a$*** *i* ***b$*** *uopšte nisu od značaja. Jedino je bitno da funkcija ima četiri parametra odgovarajućeg tipa (dva numerička i dva alfanumerička) i to onim redosledom koji smo već dali. Takođe, adresa uz funkciju USR mora odgovarati početku samog programa u memoriji.*

Izvršenje rutine SEARCH usledиće kada god, na bilo koji način, računar bude naveden da izračuna vrednost funkcije FN f. Međutim, treba imati u vidu sledeće: mašinski program je napravljen tako da posle njegovog izvršenja, a to znači posle obavljenog pretraživanja i izmena u bejzik listi, računar jednostavno zaboravlja kako je u taj mašinski program ušao. Niti će, dakle, funkcija FN f zaista biti izračunata (njen rezultat bi trebalo da bude neki broj), niti će biti dovršena naredba koja je zahtevala računanje funkcije. „Spektrum" će se ponašati kao da je uspešno obavio celu naredbu i normalno preći na sledeću. Međutim, ne mogu se sve naredbe tako prekidati na pola izvršenja, a da to ne izazove neke probleme. Nama je ovde dovoljno da postoji bar jedna naredba koja ne pravi komplikacije kad se prekine, a to je RANDOMIZE. Tako, na primer, ako želimo da u bloku između linija ***10*** *i* ***100,*** *sve naredbe* ***GO TO 1000*** *zamenimo sa* ***GO TO 2000,*** *otkucaćemo:*

RANDOMIZE FN f(10,100," GO TO 1000"," GO TO 2000")

Naravno, ne dolazi u obzir kucanje GO TO slovo po slovo, jer to računar neće prepoznati kao naredbu. Međutim, pravi GO TO simbol se ne može dobiti sa „L" kursorom. Moraćemo privremeno da pritisnemo THEN, koje ćemo kasnije lako ukloniti.

Ako se naredba za pretraživanje nalazi u okviru programa, i to baš u bloku koji se menja, promene će se desiti i u samoj toj naredbi, mada to nikako neće uticati na njeno dalje izvršenje.

Na kraju, treba nešto reći i o mogućim prekidima u toku izvršenja programa SEARCH.

Pre svega, obavlja se kontrola ulaznih podataka. Do prekida sa raportom „Q Parameter error" doći će u sledećim slučajevima:

1. Početna linija „s" je veća od konačne linije „e".

2. Niz „a$" ima dužinu nula (prazan niz).

3. Zbir dužina nizova „a$" i „b$" je veći od 243.

U svim ostalim slučajevima se pristupa pretraživanju teksta, sa zadatim parametrima. Ako linijski broj „s" ne postoji u programu, uzima se prvi sledeći, a ako ni takav ne postoji, smatraće se da je naredba za pretraživanje uspešno izvršena — sledi raport „0 OK". Inače, tekst se pretražuje liniju po liniju, sve dok linijski brojevi ne postanu veći od „e", ili se u međuvremenu ne stigne do kraja programa. Svaka linija se prvo iskopira u editni prostor, a onda se analizira. Kad god se naiđe na niz „a$", on se izbacuje, a umesto njega se ubacuje niz „b$", ili se ne ubacuje ništa ako je niz „b$" prazan. Zatim se proverava sintaksa nove linije. Prekid sa raportom „C Nonsense in BASIC" nastupa ako:

```
 10 REM +--------------------+
 20 REM !  Program: "SEARCH"  !
 30 REM +--------------------+
 40 REM
 50 CLEAR 29999
 60 LET a=30000
 70 FOR n=0 TO 6: PRINT n+1
 80 READ k,a$: LET s=LEN a$
 90 FOR t=1 TO s-1 STEP 2
100 LET h=CODE a$(t)-48
110 LET l=CODE a$(t+1)-48
120 IF h>9 THEN  LET h=h-39
130 IF l>9 THEN  LET l=l-39
140 LET b=16*h+l
150 POKE a,b
160 LET s=s+b
170 LET a=a+1
180 NEXT t
190 IF s<>k THEN  PRINT "ERROR
in line ";1000+10*n: STOP
200 BEEP 0.2,24: CLS : NEXT n
210 SAVE "search"CODE 30000,440
220 PRINT "VERIFY": VERIFY ""CO
DE 30000,440
1000 DATA 8078,"dd2a0b5cdd5e04dd
5605ed53fd5bdd6e0cdd660d22fb5ba7
ed32da8b28dd6e1fdd662022f75bdd4e
16dd4617ed43f55b78b1ca8b280911f4
00ed52d28b28dd6e"
1010 DATA 7595,"14dd661511005bed
b0ed53f95bdd6e1ddd661edd4e1f78b1
2802edb02a515ce53a475cf52a455ce5
3effcd01162afd5bcd6e19e57efe4030
3b46234e2afb5ba7"
1020 DATA 6562,"ed42387bed43455c
cd541f3076cdb016d1fdcbc5862a495c
e521000022495ceb22fd5bcd5518e122
495c2a595c232323180718bf18492af3
5b11005b237efe0d"
1030 DATA 8272,"28431abe20f622f3
5bfd46bb1abe20e6231310f8fdcbc5c6
fd360d012af35bed4bf55bcde8192b3a
f75ba728cc234fc5cd5516eb132af95b
c1edb01beb18ba18"
1040 DATA 7073,"7818af1872fdcbc5
4628632a595c23232323225d5cfdcb01
becd1e1bfdcb01fefdcb007eca8a1cfd
360d01ed5b595c131313132a615c37ed
527d3db4ca8a1ce5"
1050 DATA 6372,"2afd5b23234e2346
d1d5e5eba7ed42c5444dd4051fc1e1cd
e819c1c5cd5516132a615c2b2bc1c5ed
b8ebc1702b712afd5bcdb819eb188a18
29e1e122455ccb7c"
1060 DATA 6721,"202ccc6e19cdb819
ed53555c2bd1fd720d141e00cd8b19e1
cd1516ed7b3d5cc3761be1e122455cf1
32475cc35205d1fd720de1cd1516cfff
```

1. Nova linija ne sadrži ni jedan karakter.

2. Sintaksa nove linije nije korektna.

U oba slučaja zadržava se stara linija.

Konačno, prekid „4 Out of memory" nastupa u slučaju da:

1. Nedostaje memorija da bi se linija kopirala u editni prostor.

2. Nedostaje memorija da bi se u liniju umetnuo niz „b$"

3. Nedostaje memorija u fazi sintaksne provere nove linije.

Ponovo se u svakom od ovih slučajeva zadržava stara linija.

U slučaju dužih programskih blokova, proces pretraživanja može i da potraje. Međutim, program uvek možemo prekinuti uobičajenom komandom BREAK. Do zaustavljanja sa raportom „D BREAK —CONT repeats" doći će čim se kompletira započeta linija.

Treba još dodati i jednu napomenu u vezi sa linijskim brojevima „s" i „e". Računar uopšte ne proverava da li je neki od njih veći od 9999. Čak ne proverava ni to da li su oba broja cela, mada ih tretira isključivo kao cele brojeve. Ako, nekim slučajem, unesemo parametre „s" i „e" kao decimalne brojeve, ili kao aritmetičke izraze sa necelobrojnim rezultatom, računar će te vrednosti pogrešno pročitati.

Jovan Skuljan

SPEKTRUM

PRIKAZ TEKSTA

Pisanje programa je lepa i zabavna stvar. Međutim, kada završimo neki program i poželimo da ga damo nekome na korišćenje, postavlja se pitanje: šta sa uputstvima? Možemo, naravno, da ih napišemo na hartiji, ali je mnogo elegantnije da sam program, na zahtev korisnika, prikaže uputstva. Na žalost, tu tek počinju problemi. Napisati program koji lepo i dopadljivo prikazuje uputstva nije ni malo jednostavan posao (pogotovo ne na „spektrumu"), a uz to je i prilično dosadno ponavljati ga za svaki novi program. Ova mašinska rutina bi trebalo da reši stvar. Uputstva se pozivaju sa RANDOMIZE USR, a povratak u glavni program vrši sa S (STOP).

*Otkucajte program i snimite ga sa: SAVE „TEKST" CODE 60000,973. Program je potpuno relokatibilan, dakle možete ga učitati bilo gde u memoriju. Na adrese XXXX+825 i xxxx+826 treba poukovati koliko linija od po 32 slova ima tekst koji treba prikazati (minimum 16, inače program gubi smisao), a na XXXX+827 i XXXX+828 treba poukovati adresu teksta. Posle toga sledi RANDOMIZE USR XXXX; XXXX je, naravno, adresa na koju je učitan program. Evo kako sve to izgleda na promeru. Neka se tekst koji treba prikazati nalazi na adresi 50000 i neka je dugačak 1600 bajta, dakle 50 linija. (Ako vas mrzi da kucate toliki tekst, otkucajte sledeću liniju: FOR I=50000 TO 51599: POKE I, INT(RND*50+32): NEXT I). Program učitajte na 40000 i kucajte:*

*POKE 40825,50−256*INT(50/256)*
POKE 40826, INT (50/256)
*POKE 40827,50000−256*INT(50000/256)*
POKE 40828, INT(50000/256)

Vladimir Kostić

```
 10REM CTRL CODE HEX LOADER

 20 BORDER 1: PAPER 1: INK 7: P
OKE 23609,20
 30 CLEAR 26999: POKE 23658,8:
PRINT AT 20,0
 40 INPUT "START ADDR:";A
 50INPUT (A);": ";LINE A$:IF A$
=""THEN STOP
 60 LET C=0: IF LEN A$<>18 THEN
 GO TO 90
 70 FOR I=1 TO 17 STEP 2: LET X
=16*CODE A$(I)-768-112*(A$(I)>"9
")+CODE A$(I+1)-48-7*(A$(I+1)>"9
"): IF I<17 THEN  POKE A,X: LET
A=A+1: LET C=C+X
 80 NEXT I: LET A=A-8: IF C-256
*INT (C/256)=X THEN  PRINT A;":
";A$: LET A=A+8: GO TO 50
 90 PRINT #1; FLASH 1;"ERROR!":
 BEEP 1,1: GO TO 50
```

```
PRIKAZ TEKSTA
CTRL CODE HEX LIST

60000: DDE5213D0309E5DDEE
60008: E13E24DD5E00DD56B1
60016: 01DD6E02DD66030990
60024: EB09732372DD23DDD9
60032: 23DD23DD233D20E363
60040: DDE12A99ED7CB5C867
60048: 3A6A5C3268ED3E08CD
60056: 326A5CCD6B0D21C01E
60064: 54062036FF2310FBDD
60072: 3E01CD0116216DED9E
60080: 062CCD3EEB3E02CD35
60088: 01162100002269EDB0
60096: 060FCD48EB2823C525
60104: CDF2EAC110F421008F
60112: 00226BEDAF32085CBF
60120: FF3A085CFE36CC0FAC
60128: E8FE37CCF2EAFE5319
60136: 20EA3A68ED326A5C91
60144: 1862CD48EB285DEDEC
60152: 5A232269ED216BED6E
60160: 34CD5EEB3E03328946
60168: 5C2A69ED2B181A2A63
60176: 6BED7CB5283E2B223C
60184: 6BED2169ED35CD9667
60192: EB3E1832895C2A6BED
60200: ED3E2132865C2929B4
60208: 292929ED5B9BED1964
60216: 0620CD3EE8C97E2386
60224: C5E5D7E1C110F7C9F3
60232: 2A69EDED5B99EDED3B
60240: 527CB5C911640021E2
60248: 2602CDB503C911EA71
60256: EB2102EC060EC53E11
60264: 06E5D5ED5371EBED49
60272: 5B00002277EB2A0009
60280: 00012000EDB0D1E170
60288: 131323233D20E201AC
60296: 0C00EB09EB09C110C5
60304: D5213AED1638212CBA
```

```
60312: ED1144ED060EC53E46
60320: 06E5D5ED53A9EBED81
60328: 5B000022AFEB2A0041
60336: 00012000EDB0D1E170
60344: 1B1B2B2B3D20E201CC
60352: 0C00EBED42EBED4240
60360: C110D321EAEB3E06DE
60368: E522D5EB2A00005445
60376: 5D13011F003600EDB3
60384: B0E123233D20E9C9E6
60392: 004000410042004306
60400: 004400450046004716
60408: 204020412042204386
60416: 204420452046204796
60424: 404040414042404306
60432: 404440454046404716
60440: 604060416042604386
60448: 604460456046604796
60456: 804080418042804306
60464: 804480458046804716
60472: A040A041A042A04386
60480: A044A045A046A04796
60488: C040C041C042C04306
60496: C044C045C046C04716
60504: E040E041E042E04386
60512: E044E045E046E04796
60520: 004800490048004B26
60528: 004C004D004E004F36
60536: 204820492048204BA6
60544: 204C204D204E204FB6
60552: 404840494048404B26
60560: 404C404D404E404F36
60568: 604860496048604BA6
60576: 604C604D604E604FB6
60584: 804880498048804B26
60592: 804C804D804E804F36
60600: A048A049A048AA04BA6
60608: A04CA04DA04EA04FB6
60616: C048C049C04AC04B26
60624: C04CC04DC04EC04F36
60632: E048E049E04AE04BA6
60640: E04CE04DE04EE04FB6
60648: 005000510052005346
60656: 005400550056005756
60664: 205020512052205306
60672: 205420552056205706
60680: 405040514052405346
60688: 405440554056405756
60696: 605060516052605306
60704: 605460556056605706
60712: 805080518052805346
60720: 805480558056805756
60728: A050A051A052A05306
60736: A054A055A056A05706
60744: C050C051C052C05346
60752: C054C055C056C05756
60760: E050E051E052E05306
60768: E054E055E056E05706
60776: 0000000000505245E7
60784: 53533A20140136145F
60792: 002C201401371400AC
```



```
60800: 204F52201401531450
60808: 002E2028444F574EAE
60816: 2F55502F53544F5049
60824: 2900000000B00398D
60832: 03340008034E000D9D
60840: 035300DE005E00099B
60848: 036300E80069009249
60856: 0072000B037F00AFAE
60864: 00840092008B0008A9
60872: 039300E8009B000922
60880: 039E000B03A200FE4F
60888: 00AA000903B0000B71
60896: 03B8000B03BB00098D
60904: 03BF003601C7000BCB
60912: 03D5003B03DB00DECF
60920: 00E9000903ED00391B
60928: 03FF008A010201A232
60936: 010D01110114011740
60944: 013201DA023701CC14
60952: 023A01E40245014982
60960: 014C014F016C018A95
60968: 017201750100000EA
```

Spektrum

Kako ispisivati bitove

Kompjuteri, u osnovi, rade sa binarnim brojnim sistemom, pa je često potrebno pretvoriti neki broj iz dekadnog u binarni oblik.

Program 1 to sasvim lepo radi (dali smo univerzalan program koji će raditi na svakoj mašini). Broj koji se prevodi stavlja se u N, a u X registar stavlja se 128. Sledeća naredba ispituje da li je N veća ili je jednako od X. Ako jeste, znači da je bit 7 jedinica, štampa se 1 i od N oduzme X; ako nije, štampa se 0. Zatim se X podeli sa dva, dakle X je 64, i ceo ciklus se ponavlja ukupno osam puta. Za broj 123, na primer, biće ispisano 01111011.

Svaki računar, pa i „Spektrum", ima puno skrivenih mogućnosti. Otkrivanje tih malih tajni je dugotrajan i ne baš lak posao, ali se višestruko isplati. Program 2 je lepa ilustracija za tu tvrdnju. Radi isto što i program 1, ali je kraći, brži, i na kraju krajeva mnogo elegantnije rešen. Osnovna ideja je sledeća: jedan deo memorije (tačno 6144 bajta) se na ekranu televizora prikazuje kao slika. Pri tome svaki bit odgovara jednoj tačkici na ekranu. Postoji i naredba POINT koja ispituje da li je neka tačkica osvetljena ili nije, drugim rečima, ispituje bitove u video memoriji. Zašto to ne iskoristiti?

Program je krajnje jednostavan. Uneti broj N se „pokuje" na adresu 22432 (to nije adresa početka VIDEO memorije već adresa početka kordinantnog sistema), a zatim se pomoću ciklusa koji ide od 0 do 7 ispituju pojedini bitovi. Pri tome POINT (0,0) ispituje sedmi bit, POINT (1,0), šesti ,POINT (2,0) peti, i tako redom, sve do nultog bita.

Ako je potrebno prevesti šesnaestobitni broj, onda u programu 1 broj 128 u liniji 20 treba zameniti sa 32768, a u programu 2 linije 10 i 20 sa: 10 INPUT N: POKE 22432, INT (N/256): POKE 22433, N-245*INT (N/256) 20 FOR I=O TO 15

Vladimir Kostić

```
 0>REM PROGRAM 1

10 INPUT N
20 LET X=128
30 IF N>=X THEN  GO TO 60
40 PRINT 0;
50 GO TO 80
60 PRINT 1;
70 LET N=N-X
80 LET X=X/2
90 IF X>=1 THEN  GO TO 30
```

```
 0>REM PROGRAM 2

10 INPUT N: POKE 22432,N
20 FOR I=0 TO 7
30 PRINT POINT (I,0);
40 NEXT I
```

Male tajne velikih majstora programiranja

Trikovi na „komodoru"

Vertikalna tabulacija *je moguća na „komodoru 64" uz pomoć instrukcije:*

POKE 214, Y: PRINT „(cursor dole) VAŠ TEKST", gde Y predstavlja broj linije na kojoj treba štampati tekst.

Kombinovana sa TAB, ova POKE instrukcija može da zameni instrukciju PRINT AT koju poseduju neki drugi računari:

POKE 214,Y: PRINT TAB (X) „(cursor dole) VAŠ TEKST", gde X predstavlja broj kolone od koje treba štampati tekst.

Kada uključite „komodor 64", neki tasteri imaju automatsko ponavljanje ***(autorepeart)*** *dok drugi nemaju. Postoje, u stvari, tri mogućnosti vezane za vrednost registra 650 (decimalno):*

1. 0—63: stanje kad uključite računar;

2. 64—127: nijedan taster nema automatsko ponavljanje;

3. 128—255: svi tasteri imaju autorepeat.

Ove pojedinosti pokazuju da su važna stanja bitova najveće težine adrese 650. Ako je stanje bitova 6 i 7 nula, funkcionisanje je normalno. Ako je samo bit 6 postavljen na 1, nijedan taster ne poseduje autorepeat. Ako je bit 7 postavljen na 1, svi tasteri imaju mogućnost automatskog ponavljanja.

Da biste ***onemogućili štampanje*** *na ekranu listinga nekog bejzik programa, dovoljno je da tom programu dodate sledeću liniju:*

10 REM (shift/L)

Pazite na (shift/L) predstavlja znak koji se dobija pritiskom na dva tastera istovremeno.Prilikom pokušaja listanja programa, računar će štampati:

10 REM

? SYNTAX ERROR

Možete, naravno, izabrati neki drugi broj za ovu liniju; najpametnije bi bilo staviti ovo REM za zaštitu na početak programa.

Drugi način zaštite bejzik programa bila bi i instrukcija POKE 775,200. POKE 775,167 neutrališe ovu zaštitu. Još efikasnije (za one koji ne znaju trik . . .):

POKE 808,225: POKE 818,32
koja ne samo da zabranjuje listanje programa već i onemogućava korišćenje instrukcije SAVE. Pokušaj snimanja programa na kasetu bi doveo do blokiranja računara. POKE 808,237: POKE 818,237 vraća mašinu u normalno stanje.

Resetovanje „komodora 64" koje izaziva razne reinicijalizacije, sa gubljenjem podataka iz memorije (programa i promenljivih), može se dobiti instrukcijom SYS 58260. Ovo nije tako efikasno kao korišćenje instrukcije SYS 64738, koja je poznatija i vrši totalnu reinicijalizaciju sistema.

Vertikalno **„skrolovanje"** ekrana (kretanje linija prema gore) lako se dobija instrukcijom SYS 59626. Poziv ove rutine pomera ekran prema gore za jednu liniju. Uključivanje ove instrukcije u petlju omogućava vam da „skrolujete" po želji.

Bilo koju liniju ekrana možete **izbrisati** instrukcijom POKE 781,Y: SYS 59903, gde Y (0–24) predstavlja broj linije.

(Prema L'ordinateur individuel)

Đorđe Janković

Komodor 64

Beepkey

Ovaj jednostavan program, dugačak svega 99 bajtova, služi kao zvučna indikacija da je neki taster pritisnut. Koristi se prvenstveno kod ukucavanja dužih programa, kada nije zgodno gledati u časopis, tastaturu i ekran u isto vreme. Kada pritisnemo bilo koji taster (izuzetak C=,SHIFT, CRTL i RESTORE) čućemo zvučni signal. Ako ovo nekome smeta, potrebno je nakon startovanja programa otkucati: POKE 650,0 ili POKE 650,128.

Moguće je menjati boju tona sa POKE 49210, a, jačinu sa POKE 49205,b, kao i vreme trajanja signala: POKE 49225, a, gde su a i b brojevi između 0 i 255, odnosno 0 i 15.

```
10 C=49152
15 FORI=1TO13:S=0
20 FORJ=1TO8
25 READA$:GOSUB70:POKEC,X:C=C+1
30 S=S+X:NEXTJ
35 READP:IFS<>PTHEN60
40 NEXTI
45 PRINT"IN NEMA GRESAKA !":SYS49152
50 END
55 X=X*16
60 PRINT"IN GRESKA U LINIJI:";
65 PRINTPEEK(63)+256*PEEK(64):END
70 X=0:G=ASC(A$)
75 X=(G<64)*(48-G)+(G>64)*(55-G)
80 X=X*16
85 F=ASC(RIGHT$(A$,1))
90 X=X+(F<64)*(48-F)+(F>64)*(55-F)
95 RETURN
100 DATA A9,7F,8D,8A,02,A9,FF,85,1134
105 DATA FD,78,A9,16,8D,14,03,A9,897
110 DATA C0,8D,15,03,58,60,A6,C5,904
115 DATA E0,40,F0,09,A6,FD,E0,00,1180
120 DATA F0,07,4C,2C,C0,A2,FF,86,1110
125 DATA FD,4C,31,EA,A9,0F,8D,06,943
130 DATA D4,8D,05,D4,A9,0F,8D,18,919
135 DATA D4,A9,AA,8D,01,D4,A9,AB,1245
140 DATA 8D,00,D4,A9,11,8D,04,D4,896
145 DATA A0,32,A2,00,CA,D0,FD,88,1171
150 DATA D0,F8,A9,20,8D,04,D4,A9,1183
155 DATA 00,8D,18,D4,A2,00,86,FD,926
160 DATA 4C,31,EA,00,00,00,00,00,359
```

Pošto se program koristi kao deo INTERRUPT REQUEST rutine, on nešto usporava rad bejzika. Isključuje se istovremenim pritiskom na tastere RUN/STOP I RESTORE ili sa POKE 50000,0:SYS50000, a ponovo uključuje sa SYS 49152.

Program je moguće koristiti u kombinaciji sa druim mašinskim programima, pri čemu treba paziti da se ne koristi adresa $FD.

Amstrad

Rožete

Sledeći program je vizuelno dosta interesantan, a posebno za kompjutere sa većom grafičkom rezolucijom. Izborom dva parametra, A i B, bira se oblik krive koju kompjuter crta. Kao što je dato u programu, broj A mora biti ceo, a B ceo i uz to pozitivan. Treći parametar, K, određuje samo broj tačaka koji je K*B i ne utiče na oblik krive. Ako izaberete varijantu u kojoj se ne ucrtavaju dijagonale između tačaka, bolje je uzeti veći broj K, iako ćete za izračunavanje morati malo da sačekate — oko 1 minut. Ako izaberete varijantu sa ucrtavanjem dijagonala, najlepše slike ćete dobiti kada je K*B između 15 i 40. Program barata, u stvari, sa parametrom M, koji je jednak A/B. Obratite pažnju na sledeće oblasti u kojima se kreće parametar M:

$M \leq 1$	kriva je pericikloida
$-1 \leq M \leq 1/2$	hipocikloida
$-1/2 \leq M \leq 0$	hipocikloida
$0 \leq M \leq 1/2$	epicikloida
$1/2 \leq M \leq 1$	epicikloida
$1 \leq M$	pericikloida

Ivan Sajić

```
 5 mode 2
 9 print „program rozete"
10 print „ovaj program može da vam demonstrira
   grafičke mogućnosti vašeg kompjutera"
20 print „brojevi K i B koje je potrebno zadati
   kompjuteru moraju biti pozitivni, a A može biti i
   negativan, i sva tri moraju biti celi".
30 print „broj tačaka sa kojima se radi je K*B i
   mora biti manji od hiljadu."
40 dim X(1000), y (1000)
50 input „k=";k
70 input „a=";a
80 input „b=";b
90 m=a/b
100 n=k*b
110 print „broj tačaka je ";n
120 print „da li da se ucrtavaju dijagonale?";
130 input o$:if o$>< „da" and o$>< „ne" then 130
140 p=319/(abs(1+m)+abs(m)/
150 q=199/(abs(1+m)+abs(m)/
160 for j=o to n-1
170 i=j*b*2*pi/n
180 x(j)=320+p*((1+m)*cos(m*i)+m*cos((1+m)*i))
190 y(j)=200+q*((1+m)*cos(m*i)+m*cos(1+m)*i))
200 next j
220 for i=o to n-1
230 plot x(i), y(i)
240 next i
250 if o$=„ne" then 50
260 for i=1 to n-1
270 for j=o to i-1
280 plot x(i), y(i)
290 draw x(j), y(j)
300 next j
310 next i
320 goto 50
```

Računar „galaksija"

Ziul

Zuil je prva ozbiljnija avanturistička igra pisana za „galaksiju". Igara avantura, jasno, ima različitih, ali su se kroz kratku istoriju razvoja računara pojavila dva osnovna tipa — u jednima je cilj rešiti neke probleme i ispuniti zadatak posle čega igra postaje potpuno neinteresantna. Kod drugih su pravila igre manje više unapred poznata, dok je značajna uloga dodeljena generatoru slučajnih brojeva. Posle prvih par partija upoznajete tajne igre i ona tek onda postaje pravi izazov — možete li da je uspešno završite u što manjem broju poteza ili da postignete što veći skor? ZIUL pripada ovoj drugoj vrsti igara, što znači da ćemo vam, uz uputstvo za unošenje programa, saopštiti i čitav scenario igre.

Glavni deo ZIUL-a je pisan na bejziku dok je jedan potprogram napisan na mašincu radi uštede vremena i prostora, posebno ovoga drugoga — ZIUL, naime, zauzima 6 kilobajta „galaksijinog" RAM-a praktično do poslednjeg bajta. Da biste ga uneli najpre otkucajte NEW 200 a zatim, uz pomoć UTM-a ili nekog sličnog programa, unesite heksadekadni sadržaj sa listinga 1. Ponovo otkucajte NEW 200 i otkucajte čitav bejzik program 2. Bitno je da, osim u slučaju da imate memorijsko proširenje, skraćujete naredbe na jedno slovo kako je učinjeno u listingu, jer će samo na taj način program stati u memoriju. Nema, naravno, nikakve prepreke da koristite naša slova Č, Ć, Ž i Š koja u listingu nisu zastupljena zbog karakteristika štampača. Po upisivanju programa možete da ga snimite na traku sa SAVE i da počnete sa igrom.

Potrebno je da upravljate patuljkom Ziulom koji se neočekivano našao u lavirintu, domu strašnog carobnjaka Marduka. Marduk, na svu sreću, nije kod kuće ali mnoštvo zlih duhova koji žive u lavirintu sa nestrpljenjem očekuje njegov povratak svakoga trenutka. Jedini način da Ziul izvuče živu glavu je da nađe skriveni prsten koji će, kada mu to bude naređeno, pokazati put ka izlazu. Marduku je, naravno, vrlo stalo da tako dragocenog prstena, pa ga je dobro sakrio maskiravši ga u krivinu. On je, međutim, naprasno i na brzinu napustio lavirint, pa maska nije savršena — kada Ziul nagazi na lažnu krivinu biće upućen u sledeću sobu u smeru kretanja, a ne u fiksnu sobu kao kod ostalih krivina. Ziul može da nađe prsten jedino ako pronađe način da uđe u maskiranu sobu, u čemu će mu, možda, pomoći amuleti, koje ne treba podcenjivati zbog njihovih smešnih imena. Po lavirintu je sakriveno i dosta novca koji može da posluži za potplaćivanje zlih duhova (svuda ta korupcija), a ne sumnjamo da će Ziul naći dobru upotrebu i za dukate koji mu preostanu po izlasku iz lavirinta.

Ziul razume sledeće komande (uvek možete da kucate punu reč ili samo njeno prvo slovo):

NORTH	ili	N
SOUTH	ili	S
EAST	ili	E
WEST	ili	W
MAPA	ili	M
FENJER	ili	F
LAMPA	ili	L
BORBA	ili	B
TELEPORT	ili	T
KRAJ	ili	K.

Evo, na kraju, i nekoliko završnih napomena koje će koristiti svima koji se u početku igre „zaglibe" (nemojte ih čitati ako sami volite da otkrivate tajne avantura):

** Ziul je preslab za borbu ukoliko ne poseduje amulet.*

** Amuleti se čestom upotrebom troše.*

** Zaliha sveća za fenjer nije neograničena.*

** Đavoli ne vole svetlost pošto se boje da će se na svetlosti videti njihovo blago.*

** Teleportovanje se ne preporučuje cicijama.*

Dejan Ristanović

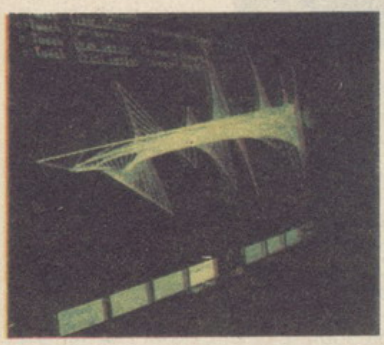
```
 90 A=U.(&2C3A):B.&2BB2,10:A.(29
)
 100 T.0:F.I=0TO16:T.X$(I):N.I:F
.I=20TO29:T.X$(I):N.I
 110 H.:P."      OTKUCAJ NIVO IGR
E:":P."(1 = POCETNIK, 6 = MAJSTO
R)";:I.N:G=0
 120 IF(N<1)+(N>6)G.110:E.H=0
 125 M=110+I.(R.*(12-N)):S=10
 130 F.I=1TO9:F.J=1TO9:MAT(I,J)=
100:N.J:N.I:F.I=0TO9:MAT(0,I)=0:
N.I
 140 A=103:C.1000:A=101:C.1000
 150 F.I=1TO18+R.*N
 160 A=I.(RND*89+11):IFA/10=I.(A
/10)G.160:E.C.1000:N.I:U=A
 170 F.I=1TO7:A=I*100+100:C.1000
:N.I
 180 F.A=1TON+4:C.1000:N.A:A=104
:C.1000
 190 R=I.(R.*(N+4)+1)
 200 H.:P."OK, ZIUL, STIGAO SI U
LAVIRINT"
 210 P."NALAZIS SE U SOBI";Y;X
 220 P.:P."STA HOCES DA URADIS";
:I.X$:P.
 230 F.I=0TO9:IFEQX$,X$(I)G.1000
*I+2000:E.IFB.(P.X$)=B.(P.X$(I))
G.1000*I+2000
 240 N.I:P."ZIULI SU GLUPI!?"
 260 M=M-1:IFM>0G.220
 300 P.:IFH=2E.P."VRATIO SE GLAD
NI CAROBNJAK,":IFH=1H=2:P."ALI T
E PRSTEN STITI":G.220:E.G.2220
 310 P."NEKI ZIULI ZNAJU DA PREK
INU;":P."OSTALE POJEDE CAROBNJAK
.":G.2230
 1000 X=I.(R.*9+1):Y=I.(R.*9+1):
IFMAT(X,Y)=100E.G.1000
 1010 IFA=I.(10*Y+X)G.1000:E.MAT
(X,Y)=A:R.
 1100 #"NORTH","SOUTH","EAST","W
EST","MAPA","LAMPA","BORBA","KRA
J","TELEPORT","FENJER"
 1110 #"PRLJAVO ZUTO","MEKA STEN
A","SJAJNA ZICA","OPALNO OKO","P
LAVI PLAMEN","PALANTIR","SILMARI
L"
 1120 #"SCHLOCK","YNGVI","GOLLUM
","MARDUK","LOKI","AZATOTH","DAG
ON","ABRACAX","SAURON","SATAN"
 2000 Z=-1:T=0
 2002 X=X+Z:Y=Y+T:IFX<1X=9:E.IFX
>9X=1:E.IFY<1Y=9:E.IFY>9Y=1
 2010 P."DOSAO SI U SOBU ";Y;X
 2020 A=MAT(X,Y):IFA<0A=-A
 2021 IFA=103MAT(X,Y)=-A:G.2002
 2025 IFA<200G.2100
 2030 P."TU JE ";X$(8+A/100)
 2040 MAT(0,A/100-2)=2+I.(R.*2)
 2050 MAT(X,Y)=104:A=104:G.260
 2100 IFA=101P."I NASAO SVECE":S
=S+10:G.2050
 2110 IFA=104P."I NAISAO NA SVOJ
E TRAGOVE!":G.260
 2120 IFA>10G.2310
 2125 IFA=RP."I TU NASAO CAROBNI
STAP!":R=10:G.2050
 2130 P."OH, NE!":P."TU JE ";X$(
A+19):I=0
 2140 IFMAT(0,I)>0P."ALI TE ";X$
(10+I);" STITI!":MAT(X,Y)=-A:G.2
60
 2150 I=I+1:IFI<7 G.2140
 2160 P.X$(A+19);" JE VEOMA PODM
ITLJIV;"
 2170 P."KOLKO DUKATA OCES DA MU
DAS";:I.P:IFP>GP.:P.X$(A+19);"
UVEK JEDE VARALICE,":G.2220
 2180 IFP<10+R.*5P.:P.X$(A+19);"
TE JE SMRVIO!":G.2230
 2190 P.X$(A+19);" TE OSTAVLJA N
A MIRU -":P."...ZA SADA!":G=G-P:
MAT(X,Y)=-A:G.260
 2220 P."PA TE JE POJEO!"
 2230 P.:P."I TAKO SI IZGUBIO!"
 2240 P.:P."OCES OPET";:I.X$:IFB
.(P.X$)=68P.:P."NEKI ZIULI NIKAD
A NE UCE!":F.I=1TO1000:N.I:G.100
 2250 P."DOBRO DA SI SE OPAMETIO
":S.
 2310 IFA<100MAT(X,Y)=-A:Y=I.(I.
(A/10)+.5):X=I.(10*(A/10-Y)+.5):
G.2010
 2320 P=I.(R.*5+1):G=G+P:P."I TU
NASAO";P;" ZLATNIKA":P."IMAS";G
;" ZLATNIKA":G.2050
 3000 Z=1:T=0:G.2002
 3010 G.2010
 4000 Z=0:T=1:G.2002
 5000 Z=0:T=-1:G.2002
 5010 G.2010
 6000 P=U.(14):H.:F.I=1TO9:F.J=1
TO9
 6010 B=MAT(I,J):IFB<0B=-B
 6020 IFB=104P."  T";:G.6500
 6030 IFMAT(I,J)>-1P." ??";:G.65
00
 6035 IFB=100P." $$";:G.6500
 6040 IFB>10G.6100E.P." D";
 6050 B.W.(&2A68),B+47:W.&2A68,W
.(&2A68)+1:G.6500
 6100 IFB>199B=B/100-1:P." A";:G
.6050
 6110 IFB=101P."  S";:G.6500
 6120 IFB=103B=U
 6130 P.B;
 6500 N.J:P.:N.I:P=U.(22):G.210
 7000 P."NA KOJU STRANU GLEDAS";
:I.X$:Z=0:T=0:B=B.(P.X$)
 7010 IFB=78Z=-1:E.IFB=83Z=1:E.I
FB=69T=1:E.IFB=87T=-1:E.P." SVAS
TA":G.7000
 7020 I=X+Z:J=Y+T:IFI=0I=9:E.IFI
=10I=1:E.IFJ=0J=9:E.IFJ=10J=1
 7022 B=MAT(I,J):IFB<0B=-B
 7025 MAT(I,J)=-B:IFB<10P.:P."PR
OBUDIO SI DJAVOLA!":P."I ON TE J
E POVUKAO...":X=I:Y=J:G.2010
 7030 P."VIDIS ";:IFB>199P."AMUL
ET":E.IFB=101P."SVECE":E.IFB=100
P."PARE":E.IFB=104P."SVOJ TRAG":
E.P."KRIVINU"
 7040 G.260
 8000 IFA<11E.P."PRVO NADJI DJAV
OLA PA SE BORI!":G.260
 8005 P."OTKUCAJ BROJ AMAJLIJE K
OJOM BI":P."DA NAPADAS";:I.P:IFM
AT(0,P)<1P." DA - SAMO STO JE NE
MAS":G.260
 8010 P.X$(P+10);" NAPADA I":P.X
$(A+19);" STRADA OSTAVLJAJUCI":M
AT(0,P)=MAT(0,P)-1
 8015 P=I.(RND*40+10):P.P;" ZLAT
NIKA.":G=G+P:G.2050
 9000 IFH=0P.:P."IGRU SI,NARAVNO
,IZGUBIO!":E.P.:P."IGRU SI (ZACU
DO) DOBIO!!!":P."I OSTAJE TI";G;
" DUKATA"
 9020 G.2240
 10000 IFR<10P."LUDACKI JE TELEP
ORTOVATI SE BEZ":P."CAROBNOG STA
PA!!!":G.260
 10005 P."GDE HOCES DA SE TELEPO
RTUJES?":P."X KOORDINATA";:I.Y:P
."Y KOORDINATA";:I.X:G=0:F.I=0TO
6:MAT(0,I)=0:N.I
 10010 IF(MAT(X,Y)=103)+(MAT(X,Y
)=-103)P."***** PRSTEN *****":H=
1:G.2050
 10020 G.2010
 11000 S=S-1:IFS<0P."NEMAS SVECA
":G.260:E.F.I=X-1TOX+1:F.J=Y-1TO
Y+1:IF(I<1)+(I>9)G.6500
 11010 IF(J<1)+(J>9)G.6500:E.B=M
AT(I,J):IFB<0B=-B
 11020 MAT(I,J)=-B:G.6020
```

```
&2C3A: 3E 0C E7 21 A9 28 CD 6D
&2C42: 2C 21 76 2C 11 C9 28 01
&2C4A: 0D 00 ED B0 21 E9 28 CD
&2C52: 6D 2C 11 27 29 21 83 2C
&2C5A: 01 10 00 ED B0 3E C3 32
&2C62: A9 2B 21 94 2C 22 AA 2B
&2C6A: C3 F5 0C 06 0D 3E 2A 77
&2C72: 23 10 FA C9 2A 20 20 5A
&2C7A: 20 49 20 55 20 4C 20 20
&2C82: 2A 44 2E 20 52 49 53 54
&2C8A: 41 4E 4F 56 49 5C 20 38
&2C92: 34 2E E3 D5 11 77 07 D7
&2C9A: 28 0B 11 5B 07 D7 D1 28
&2CA2: 0B E3 C3 C2 2C D1 21 BA
&2CAA: 2C C3 9A 03 21 B3 2C C3
&2CB2: 9A 03 4D 41 54 AC C3 AC
&2CBA: C2 4D 41 54 AC CC AC C2
&2CC2: C9 F1 CD D3 2C AF C3 60
&2CCA: 07 F7 F1 CD D3 2C C3 45
&2CD2: 0A EF 3A B2 2B F5 6F CD
&2CDA: BC 0A DF 28 0F CF CD BC
&2CE2: 0A CD E6 0A CD 05 00 F1
&2CEA: 3D BD 30 03 C3 8F 07 CD
&2CF2: BC 0A CD 32 0B CD 6D 0A
&2CFA: C3 37 01 C3 01 00 21 0D
```

tako će govoriti računari

Sedam načina komuniciranja sa kompjuterom

Palice, „miš", optičko pero, senzorski ekran, sistem za prepoznavanje glasa i drugi domišljati uređaji olakšavaju razgovor sa ličnim računarom. Krajem ove decenije korisnik će moći da se obrati kompjuteru rečima: „Hoću da oformim svoju dokumentaciju". A računar će odgovoriti: „U redu, pričaj mi o tome".



GRAFIKA NA DODIR

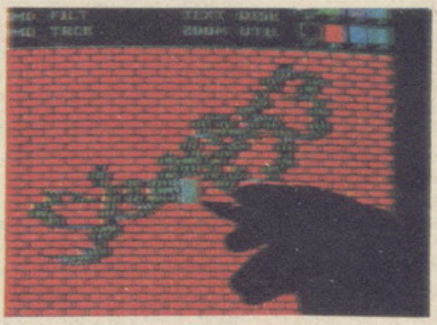
1. Dodir prstom stvara na sezorskom ekranu paučinasti crtež. Specijalni grafički program povezuje tačke koje prst dodirne, formirajući mrežu linija. Umetnik kontroliše program dodirujući okvire u dnu ekrana koji predstavljaju naredbe.

JAGODA SA GRAFIČKE TABLE

2. Plastično pero na površini osetljivoj na pritisak na grafičkoj tabli pomaže umetniku da skicira crtež. Dugme na vrhu table kontroliše mnoštvo naredbi, od izbora boje do debljine linije koja se iscrtava. Ono može, takođe da bude korišćeno za memorisanje slike u kompjuteru i njen poziv radi kasnijih promena.

BIRANJE UZ POMOĆ MIŠA

3. Klizanjem miša preko table pomera se strelica, zvana kursor, u bilo koji položaj na ekranu. Miš ima elektronske senzore koji zapisuju kretanje valjka na njegovom stomaku. Kada kursor dovede na željenu naredbu, operater pritiska dugme na mišu da bi otpočeo akciju.



CRTANJE OPTIČKIM PEROM

4. Sa optičkim perom kompjuterski operater iscrtava poruku jednako lako kao što grafiti-umetnik crta uz pomoć spreja u boji. Specijalna kola u peru saopštavaju kompjuteru u koju tačku na ekranu je pero usmereno. Kompjuter zatim iscrtava liniju koja odgovara kretanju pera.

U romanu i filmu „Odiseja u svemiru 2001" Artur Klark (Arthur Clarke) stvorio je superkompjuter nazvan „Hal 9000", koji je razgovarao na engleskom jeziku. Kada je „Hal" odbio da izvrši jednu naredbu, to nije bilo zbog toga što je nije razumeo, nego zato što se pobunio protiv svojih ljudskih gospodara.

Stvarni kompjuteri današnjice obično su sasvim voljni da izvrše zadatak, ali samo ako im se on saopšti na jeziku koji mogu da shvate. Većina računara zahteva da im se naredba otkuca na tastaturi. Sa prefinjenošću svojstvenom mašini, oni očekuju da svaka komanda sadrži prave reči, u pravilnom rasporedu, razdvojene jasnim znacima interpunkcije. Ako nedostaje samo jedan zarez, mašina može da se blokira usled neodlučnosti.

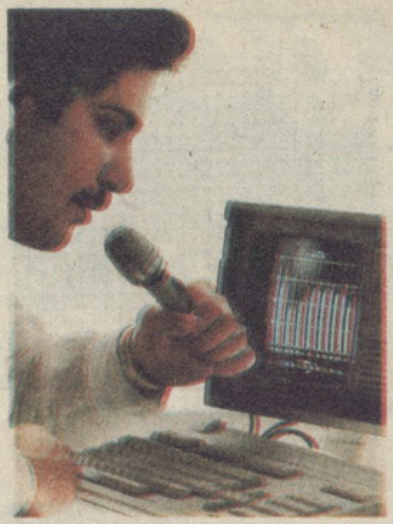
Kako zaobići tastaturu

U poslednje vreme projektanti ličnih računara počeli su mašine da prave više „prijateljskim", omogućujući da se sa njima „razgovara" uz minimalnu upotrebu tastature, i bez dobrog poznavanja veštine kucanja. Neki od ovih maštovitih novih uređaja izvedeni su iz sistema već korišćenih u većim, skupljim kompjuterima. Drugi su prilagođeni delovi vojne ili industrijske opreme, ili čak kompjuterskih akcionih igara.

Cena uređaja znatno varira — od pedeset dolara za palice, do sistema za prepoznavanje glasa koji košta najmanje 2.600 dolara. Sa ovim i drugim uređajima — „mišem", optičkim perom, grafičkom tablom i senzorskim ekranom — čak i novajlija može da „pokrene" kompjuter govorom, na primer, ili dodirom komandi ispisanih na ekranu.

Ali tastatura ipak ne može sasvim da se izostavi. Često nema drugog načina da se memoriji računara dodaju nove informacije. Projektanti kompjutera, međutim, nastoje da tastaturu zaobiđu bar kod izdavanja naredbi. Razlog: ljudi mogu da se zbune kada tastaturu koriste i za unošenje novih podataka i za saopštavanje naredbi o postupku sa tim podacima.

Klasičan primer ove zabune desio se u laboratorijama „Xerox" u palo Altou u Kaliforniji, gde je jedan istraživač koristio eksperimentalni program za obradu teksta. Misleći da u istraživački izveštaj unosi tekst, otkucao je reč „edit". Na nevolju, računar je očekivao naredbu, pa je reč protumačio kao sledeći niz komandi:



PROGRAMIRANJE PALICAMA

7. Kao što pilot koristi palice da bi upravljao avionom, operater na računaru koristi palice da bi vodio likove gore, dole i preko ekrana. Dugme na vrhu, koje se kod video igara koristi kao okidač, izvršava željenu naredbu.

DETE ZA TASTATUROM

5. Specijalna tastatura sa plastičnim poklopčićima prikazuje životinje i raznu opremu koji objašnjavaju osnove zemljoradnje. Dete pritiska kamion, na primer, i ovaj se pojavljuje na ekranu. Pritiskom na taster dete može da dobije informacije o vozilu, koje se pojavljuju odštampane ispod ekrana. Na raspolaganju su i razni drugi poklopčići.

RAZGOVOR SA RAČUNAROM

6. Operater upravlja računarom opremljenim uređajem za prepoznavanje glasa. Električna kola mašine beleže glas i pretvaraju ga u digitalni obrazac jedinica i nula. Kompjuter poredi ovaj zapis sa spiskom od 50 izgovorenih naredbi koje se nalaze u njegovoj memoriji, donosi sud o tome kojoj odgovara, i izvršava naredbu.

E(verything)	Sve
D(elete)	Izbriši
I(nsert)	Unesi tekst
T	Otkucaj T.

Kompjuter je, ukratko, izbrisao čitav sadržaj i zamenio ga jednim jedinim slovom T.

Kornjača na ekranu

Novi uređaji pomažu da se spreče ovakve nezgode. Mnogi među njima omogućuju osobi da pomeri elektronski pokazivač (pointer) do jedne ili više naredbi koje su se pojavile na ekranu, izbegavajući, na taj način, rizik tipografskih grešaka koje mogu da zbune računar. Neki izdavači softvera za obrazovanje su za vođenje pointera adaptirali palice koje se koriste za popularne video igre. Na primer, disk nazvan „Turtle-Toyland Jr." od 35 dolara nudi deci palice za kontrolu položaja pointera na ekranu — ovog puta je to jedan lik koji podseća na kornjaču. Pomerajući kornjaču od jedne ilustrovane naredbe do sledeće i pritiskajući dugme „select" na palici, dete navodi računar da sklopi program — skup naredbi koje mu saopštavaju da nešto uradi.

Jedan od uređaja za pointiranje zove se „miš", a koristi ga „Apple" za svoje računare „liza" i „mekintoš" — da bi se odabrao položaj kursora, odnosno elektronskog pointera na ekranu. Klizeći mišem preko ravne površine, kao što je površina stola, plastični valjak na dnu rotira kao kuglični ležaj. Elektronski senzori u mišu registruju kretanje valjka i prenose ga u kompjuter. Osoba koja kompjuter koristi postavlja kursor preko simbola za željenu naredbu na ekranu i pritiska dugme na mišu da bi pokrenula izvršenje komande.

Na sve privlačnijem području kompjuterske grafike, dva naročita uređaja za pointiranje primoravaju budućeg umetnika da koristi tastaturu. Jedan od njih, nazvan grafička tabla — ravna, elektro-senzitivna ploča za crtanje — može da se koristi za crtanje noktom ili perom (vidi str. 16 do 19). Žicama povezana sa kompjuterom, tabla beleži svaku tačku kontakta i trenutno prenosi sliku na ekran.

Drugi, optičko pero, potiče iz industrijskih biroa, ali je sada na raspolaganju i za kućnu upotrebu. To je instrument u obliku olovke koji, kada se prevuče preko ekrana, navodi računar da po istom tragu iscrta liniju. (U programima za projektovanje sa tastaturom operator treba da ukucava naredbe koje kompjuteru saopštavaju gde mora da iscrta koju liniju.)

Preraspodela tereta

Optička pera se zasnivaju na postupku kojim se formira lik na ekranu računara, ili bilo kom televizijskom ekranu. Snop elektrona prelazi preko ekrana 30 puta u sekundi, izazivajući svetljenje njegove fosforescentne prevlake. Pero sadrži električno kolo osetljivo na svetlost, koje detektuje prolazak ovog snopa i istog trena šalje kompjuteru signal. Računar, zatim, određuje položaj pera na osnovu pozicije snopa na ekranu kada je pristigao signal.

Na ekranu svako može da crta lako kao na papiru, pa čak i da pero koristi kao elektronsku gumicu za brisanje, vraćajući je unatrag po liniji. Direktor za projekte firme „Henry Dreyfuss Associates" Alan Frenk (Frank), jedan od tvoraca popularnog telefona i „polaroid" kamere, kaže o tome: „Optičko pero skida teret sa čoveka i prebacuje ga na kompjuter, gde ovaj i pripada".

I sama tradicionalna tastatura pretrpela je promene: mnogi kućni računari prošarani su tasterima za specijalne funkcije koji na običnoj pisaćoj mašini ne postoje. Ovi tasteri izvršavaju zadatke poput kretanja kursora ili aktiviranja često korišćenih naredbi. Jedna nova tastatura firme „Polytel Computer Products", velika 28×60 cm, ima 717 tastera koji funkcije menjaju u zavisnosti od toga koji program kompjuter izvršava. Za svaki program upotrebljavaju se drugačiji plastični poklopčići koji označavaju nove funkcije tasterâ.

Još suptilniji su senzorski ekrani koji se već koriste u centrima za lansiranje projektila i kontrolnim prostorijama elektrana, gde su delići sekunde od suštinskog značaja. Kao i sa mišom ili palicama, kompjuter prikazuje veliki broj naredbi na ekranu — koje se na primer, sastoje od naziva naredbe nacrtane u okvir. Operater, jednostavno, treba da dodirne okvir prstom i na taj način pokreće komandu.

Senzorski ekran firme „Hewlett-Packard" koristi dva skupa infracrvenih zraka, koja se projektuju iz orkužujućeg rama, ukrštena tačno ispred ekrana. Kada se prstom ili nekim predmetom dodirne ekran, par svetlosnih zraka se poremeti, slično kao što provalnikov prolazak omete alarmni uređaj osetljiv na svetlost. Kompjuter zatim ove prekinute zrake koristi kao **x** i **y** koordinate na grafikonu da bi odredio položaj prsta.

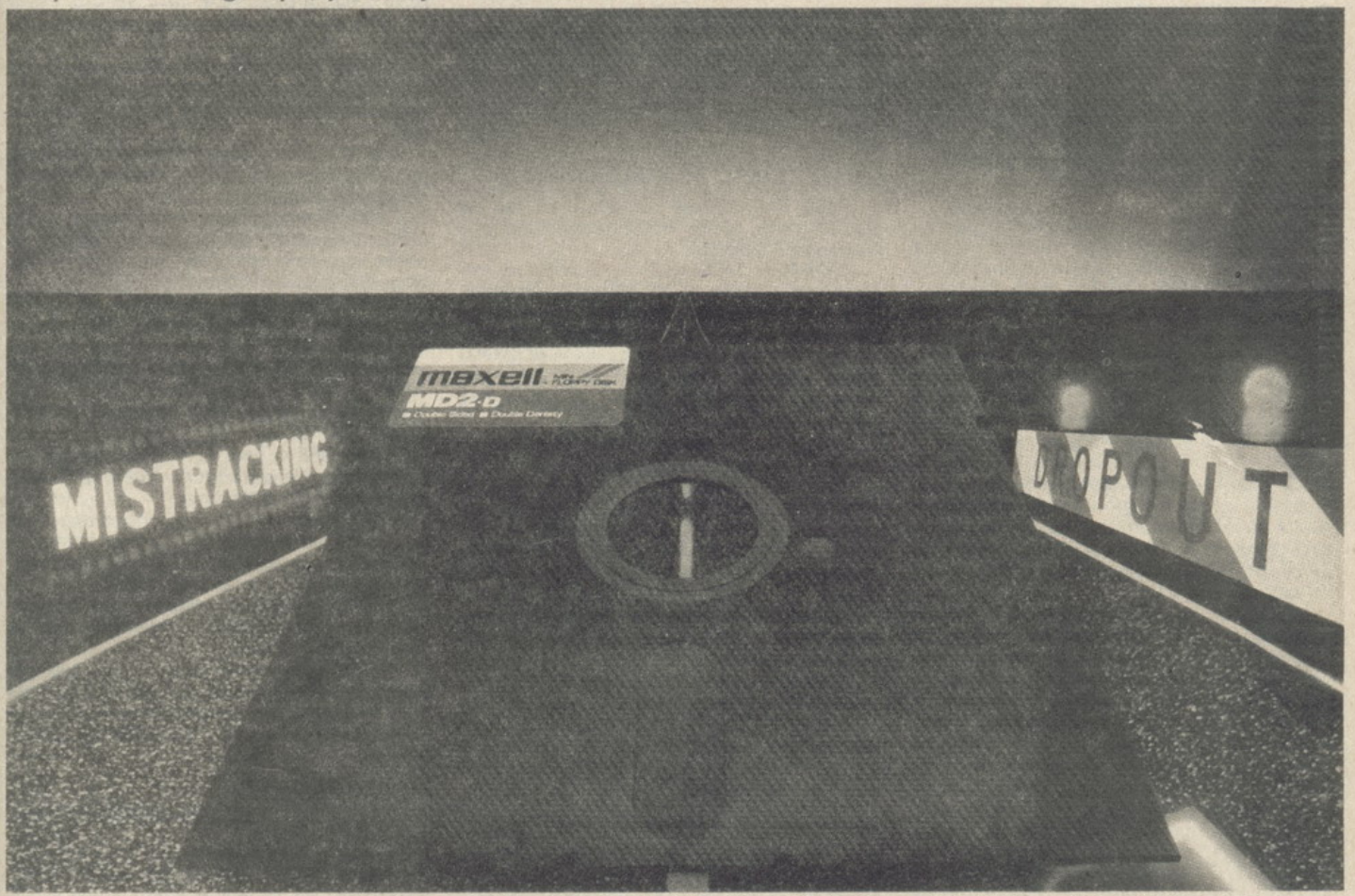
Nova komunikaciona alatka koja najviše podseća na izmaštani kompjuter „Hal" je sistem za prepoznavanje glasa koji po ceni od 2.600 dolara nudi firma „Texas Instruments". Za razliku od „Hala", ovaj sistem ne shvata doslovno ono što mu je rečeno, ali kada osoba govori u mikrofon, kompjuter prevodi njene naredbe u skup jedinica i nula koje sačinjava digitalnu sliku njegovog glasa. Zatim ovu sliku poredi sa bibliotekom od preko 50 drugih slika uskladištenih u njegovoj memoriji da bi otkrio šta je govornik rekao. Međutim, uređaj može da prepoznaje samo jednog govornika u jednom trenutku. Svaki novi govornik mora da prvo preobuči kompjuter da bi ovaj prepoznavao njegov glas.

Kompjuteri koji će doista razumevati govor, kao što je „Hal" činio u „Odiseji u svemiru", ostaju primamljiv ali neostvaren san. Međutim, neki stručnjaci smatraju da bi računari već uskoro mogli da postignu sledeću dobru stvar: razgovaraju na izgled prirodnim glasovima. „Krajem ove decenije", prognozira kompjuterski istraživač iz Njujorka Čarls Leht (Charles Lecht), „osoba bi mogla da se obrati kompjuteru sledećim rečima 'Hoću da oformim svoju dokumentaciju', a kompjuter da joj odgovori 'U redu, pričaj mi o tome'. Osoba će zatim saopštiti računaru kako bi dokumentacija trebalo da bude oformljena, i tako dalje. Kompjuter će na kraju reći 'Razumeo sam'". Tako će govoriti računari.

strogo kontrolisani diskovi

Računari u akciji

U „Računarima 3" i „Računarima 4" posvetili smo pažnju izboru disk jedinica i problemima vezanim za njihovo povezivanje sa kompjuterom. Kada rešite sve probleme ovoga tipa, preostaje vam da mirno uživate blagodeti opreme u koju ste uložili dosta novca i truda. No, da disk jedinica ne bi služila samo kao brzi kasetofon, treba naučiti dosta novih stvari!



Uz disk jednicu se, po pravilu, daje i uputstvo bez koga se ne može ni zamisliti korišćenje njenih potencijala. Uputstva za upotrebu disk jedinica su, na žalost, obično prilično kratka i lakonski pisana — početnik će se naći u čudu kada se od njega bude tražilo da zna šta je formatiranje, šta direktorijum, šta biblioteka . . . Zato je korisno na samom početku upoznati strukturu diskete i njene delove.

Logička podela

Fizički posmatrano, disketa se sastoji od 40 (80) koncentričnih tragova, pri čemu je svaki podeljen na desetak sektora. Korisnik, međutim, ne mora da zna ništa o ovoj podeli — za njega se disketa sastoji od dve strane, svaka strana se sastoji od tridesetak direktorijuma, a u svakom direktorijumu su smešteni programi koje on imenuje. Obratimo malo više pažnje na ovu takozvanu logičku podelu diskete.

Jedna strana diskete obično se naziva drajvom, što izgleda prilično čudno: do sada smo smatrali da je drajv, zapravo, jedna disk jedinica. Nekada su, međutim, disk jedinice bile jednostrane ('single side'), pa su sa računarom bila neprekidno povezana dva drajva. Onda je „izmišljena" dvostrana disk jedinica — jedan uređaj koji zamenjuje dva ranija. Da se ne bi postavljalo previše problema sa kompatibilnošću softvera i literature, usvojeno je da se prva strana diskete naziva „Drive 0", a druga — „Drive 2".

Ukoliko se pri navođenju imena programa ne navede na kojoj se strani diskete nalazi, podrazumeva se drajv nula. Ukoliko duže vreme treba da radite sa nekim drugim drajvom, otkucaćete nešto poput DRIVE 2, pa će se sve buduće operacije, ukoliko eksplicitno ne naglasite nešto drugo, izvršavati sa programima na poleđini diskete.

Na svaku stranu diskete (svaki drajv) staje po 100, 200 ili čak 360 K podataka. Vrlo je zgodno da ti podaci budu logički povezani (na primer, da svi budu igre). U tom slučaju možete i da dodelite ime čitavoj disketi. Ime diskete je bilo koja reč koja ima do 10 ili 12 slova, odnosno brojeva, i dodeljuje se naredbom koja se obično zove TITLE. Pošto otkucate TITLE IGRE, disketa je nazvana IGRE, pa ćete je ubuduće lako razlikovati i pronalaziti. Uz malo pažnje, vaša „diskoteka" može da bude savršeno sređena!

Direktorijum i biblioteka

Često se dešava da je tristotinak kilobajta previše za jednu oblast. U tom se slučaju obično odlučujete da na jednu disketu stavljate podatke nekoliko vrsta. Ukoliko, na primer, vodite evidenciju nekog kluba programera, biće vam potreban spisak članova, zatim odvojeni spiskovi članova za svaku od sekcija kluba, pristupno pismo, katalog programa . . . To je idealna situacija da dođe do zbrke: šta ako, na primer, pripremate pismo koje ćete slati članovima koji imaju „spektrum" a zatim dobijen tekst nazovete PISMO brišući tako juče napisano pismo za vlasnike „komodora"? Da bi se izbegli problemi ovoga tipa, izmišljeni su direktorijumi. Naziv svake datoteke (programa, teksta, podataka . . .) se sastoji od naziva direktorijuma (obično bilo koje slovo abecede) i imena samog programa u okviru direktorijuma. Vlasnicima „spectruma" u našem primeru može da bude dodeljen direktorijum S, a vlasnicima „komodora" direktorijum C. Posle ovakvog dodeljivanja, datoteke C.PISMO i S. PISMO će biti nezavisne jedna od druge i svakoj ćemo moći da pristupamo navođenjem njenog punog imena.

Znajući da je teško neprekidno pamtiti sa kojim direktorijumom radimo, autori DOS programa obično uvode pojam radnog direktorijuma. Na početku rada možemo, na primer, da otkucamo DIR C, pa će slovo 'C' biti proglašeno za ime radnog direktorijuma. Ukoliko docnije otkucamo SAVE „PISMO", tekući tekst će biti snimljen pod imenom C.PISMO, dok ćemo za učitavanje teksta S.PISMO morati da otkucamo LOAD „S.PISMO" ili da pređemo u direktorijum S (komandom DIR S) i otkucamo LOAD „PISMO". Ukoliko ne otkucamo komandu DIR, smatraće se da radimo sa nekim nedefinisanim direktorijumom koji se obično obeležava znakom $. Ukoliko, dakle, otkucamo SAVE „PISMO", program koji snimamo će se zvati $. PISMO.

U radu sa računarom često se aktiviraju raznorazni uslužni programi: ako radimo na bejziku, potreban nam je program za ispisivanje sadržaja svih promenljivih, pronalaženje nekog stringa i njegova eventualna zamena nekim drugim i slične stvari. Ovakve programe možemo da izdvojimo na posebnu (tzv. 'utility'), disketu koja će se najčešće nalaziti u drajvu 1 (drajv nula je, naravno, rezervisan za disketu sa programom koji razvijamo i podacima za njega). Kada nam je potreban uslužni program koji se zove XYZ, možemo da otkucamo RUN :1.XYZ i tako ga izvršimo. Primećujemo da smo morali da kucamo :1 da bi označili da se program koji izvršavamo nalazi na disku čiji je broj 1, što nije mnogo prijatno ukoliko često pozivamo uslužne programe. Problem ne možemo da rešimo tako što ćemo otkucati DRIVE 1, jer ćemo tada morati da naglašavamo da se program koji razvijamo nalazi na drajvu 0 kada god poželimo da snimimo ili učitamo neku njegovu radnu verziju. Zato ćemo otkucati LIB :1. i tako naglasiti da se uslužna biblioteka (library=biblioteka) nalazi na drajvu 1. Kada, ubuduće, otkucamo XYZ, računar će primetiti da ne postoji ni jedna bejzik naredba koja se zove XYZ i pokušaće da izvrši program iz tekuće biblioteke koji nosi to ime; tek ukoliko tog programa nema, biće prijavljena sintaksna greška. Ponekad se navodi i ime direktorijuma kao ime biblioteke (npr. LIB :1.X što znači da, pre prijavljivanja sintaksne greške, treba proveriti da li postoji odgovarajući program u direktorijumu X na drajvu 1), ali početnik ne mora o tome mnogo da razmišlja.

Neki DOS sistemi omogućavaju i definisanje 'radnog programa'. U toku razvoja nekog programa iskusni programeri neprekidno snimaju radne verzije na disketu i tako se obezbeđuju od nestanka struje ili neke druge neprijatne situacije. Višestruko kucanje imena istog programa, pogotovu ako ono ima pet-šest slova, može da bude neprijatan posao kada se, na primer, pomnoži sa faktorom deset. Zato ćemo otkucati WORK „IME", pa ćemo, umesto SAVE „IME", kucati jednostavno SAVE " ", a umesto LOAD „IME" — LOAD " " (ovako se pomalo podsećamo „starih" vremena kada smo radili sa kasetofonom). Ponegde je, čak, moguće zahtevati od računara da numeriše sukcesivno verzije programa, pa da posle IME01 snima program IME02, IME03 itd. Na taj način ćemo posle svake greške imati i prethodnu verziju programa, što je, inače, odlika velikih kompjuterskih sistema. Ukoliko ne želimo da nam diskete budu pune starih verzija, neophodno je povremeno „veliko spremanje".

Formatiranje . . .

Pre nego što počnemo da snimamo svoja programerska remek-dela na disketu, potrebno je, pre svega, da naučimo da pripremimo diskete za rad, tj. da ih formatiramo. Smisao formatiranja smo objasnili još u „Računarima 3" — sada nam ostaje samo da objasnimo tehnički deo posla.

Program za formatiranje disketa se nalazi u ROM-u ili na disketi koju smo dobili zajedno sa našom opremom. U prvom slučaju, pozivamo ga nekom komandom, a u drugom ga treba posebno učitati. Formatiranje je strahovito destruktivna operacija

BBC B

Opšti format naredbe za pristup disku je:

<komanda>:<drajv>,<direktorijum>" programa>

Ovde je <drajv> broj između 0 i 3, <direktorijum> bilo koje veliko slovo abecede ili specijalni znak '$' a <ime programa> niz od maksimalno sedam slova, brojeva i specijalnih znakova u kome ne smeju da se nalaze simboli '.',',' i ':'. džoker znaci.

*Neki elementi opšteg formata mogu da se izostave. Ukoliko ne napišemo broj drajva biće podrazumevan drajv nula ukoliko naredbom *DRIVE nije drugačije specificirano. Izostavljanje direktorijuma izaziva referenciranje direktorijuma '$' ukoliko ova konvencija nije promenjena naredbom *DIR. U okviru imena programa u nekim slučajevima mogu da se koriste džoker znaci pri čemu zvezdica zamenjuje proizvoljan broj slova a povisilica jedan znak.*

Osim standardnih komandi za rad sa kasetofonom, DFS ROM dodaje sledeće komande i naredbe:

***FORM40, *FORM80** — naredbe za formatiranje diskova na 40 odnosno 80 traka. Kod Acornovog disk interfejsa ove naredbe zamenjuju programi na uslužnoj disketi.*

***VERIFY** — verifikuje ispravnost formatirane diskete. Kod Acornovog DFS ova naredba predstavlja poziv programa sa uslužne diskete.*

***ACCESS** — omogućava zaštitu od neželjenog brisanja programa.*

***ACCESS PROGRAM** — L zaključava PROGRAM dok *ACCESS PROGRAM ponovo omogućava njegovo brisanje i modifikovanje, *ACCESS =.1 L će „zaključiti sve programe u svim direktorijumima tekućeg drajva.*

***BACKUP** m n izaziva kopiranje drajva m na drajv n. Ne kopiraju se, jasno, posebno zaštićene diskete.*

***DELETE** omogućava brisanje bilo kog nezaključanog programa.*

***DESTROY** radi isto kao *DELETE ali omogućava korišćenje džoker znakova u imena programa i, na taj način, uništavanje sadržaja većeg dela diskete.*

***ENABLE** — omogućava primenu destruktivnih naredbi kao što su *FORM, *BACKUP i *DESTROY. Podaci na disketi su tako zaštićeni od ruku nestručnog korisnika.*

***BUILD** startuje sirotinjski tekst editor. Uz njegovu pomoć možemo da prenesemo bilo koji ASCII tekst na disketu.*

***CAT D** izdaje spisak programa i datoteka na drajvu D.*

***COMPACT D** sabija programe na drajvu D kako bi se oslobodio što veći povezan prostor. Preporučljivo je koristiti ovu naredbu samo kada računar prijavi grešku 'Disc full'.*

***COPY D1 D2 <ime>** — omogućava kopiranje jednog ili više (uz džoker znake) programa sa drajva D1 na drajv D2.*

***DIR X** — specificira X za radni direktorijum.*

***DRIVE D** — specificira D za radni drajv.*

***DUMP <ime>** prikazuje na ekranu heksadekadni i ASCII sadržaj imenovane datoteke.*

***INFO <ime>** daje relevantne podatke (početna i izvršna adresa, dužina i lokacija na disku) o datoteci.*

***LIB L** omogućava imenovanje radne biblioteke u kojoj će računar pokušavati da nađe programe pre nego što prijavi grešku zbog nepostojeće naredbe.*

***OPT 4** specificira postupak sa IBOOT programom, *OPT 4,0 zahteva njegovo ignorisanje, *OPT 4 1 njegovo unošenje u memoriju, *OPT 4 2 njegovo izvršavanje sa *RUN a *OPT 4 3 sa *EXEC. IBOOT program se startuje kada korisnik pritisne SHIFT BREAK.*

***RENAME** — ime 1 ime 2 — omogućava preimenovanje jednog ili (kod nekih disk interfejsa) više programa.*

***TITLE** ime — dodeljuje ime drajvu. Ime se sastoji od najviše 12 znakova i predstavlja jedino pogodnost za korisnika, računar ga ne ispituje ni u jednoj prilici!*

***TYPE** ime izdaje datoteku sa tekstom na ekranu. U datoteci ne smeju da postoje kontrolni karakteri!*

***WIPE** omogućava selektivno brisanje sadržaja diskete pri kome računar konsultuje korisnika pre nego što obriše bilo koji nezaključani program. *ENABLE nije neophodno.*

Dejan Ristanović

koja nepovratno uništava sadržaj čitave diskete — ne postoji nikakav programerski trik koji će povratiti ono što se nalazilo na disketi koju smo preformatirali! Ukoliko se program za formatiranje nalazi u ROM-u, obično je zaštićen na neki način — računar zahteva da najpre otkucamo ENABLE a zatim FORMAT, jer se na taj način sprečava da neko ko ne zna mnogo o računaru uništi sadržaj diskete. Posle komande FORMAT, ili startovanja programa za formatiranje, stupamo u komunikaciju sa kompjuterom i odgovaramo u kome se drajvu nalazi disketa koju treba formatirati, da li želimo 40 ili 80 traka (za drajv i za disketu je pogubno formatiranje na 80 traka ukoliko je drajv predviđen samo za 40, obrnuta operacija je sasvim dopustiva i često se koristi), želimo li jednostruku ili dvostruku gustinu upisa... Posle svih ovih podataka, računar još jednom postavlja pitanje poput „O.K. to format drive 0?". To je ujedno i poslednja prilika da se predomislite — ukoliko odgovorite sa Y (od yes), ulaz u proceduru formatiranja je neminovan.

U toku formatiranja računar vrši automatsku verifikaciju. Kada se traka ispravno odazove, biva ispisan njen broj. Ako verifikacija uspe tek posle nekoliko pokušaja, uz broj trake obično biva ispisan upitnik. Ukoliko, najzad, verifikacija ne uspe i posle 10 pokušaja, računar ispisuje dva upitnika posle broja „neposlušne" trake. Neki disk kontroleri će tada zapisati broj neispravne trake u neki interni katalog, pa je neće koristiti. Naš je, međutim, savet da svakoj disketi koja se neispravno formatira (pa makar se pojavio samo jedan upitnik na jednoj strani) pokažete put do kante za smeće — iako je ovakvo ponašanje naizgled neracionalno. Diskete su danas izuzetno jeftine, a programerski rad, pa makar to bio i rad nekog početnika, izuzetno skup.

... i kopiranje disketa

Nakon formatiranja nekoliko praznih disketa, treba napraviti bar jednu kopiju osnovne, sistemske diskete. Na taj način se obezbeđujemo od gubitka ovih fundamentalno važnih programa i, ujedno, vežbamo rad sa disk jedinicama. Disketa, koju kopiramo ide u drajv 0, a disketa na koju kopiramo u drajv 1, uz komandu BACKUP 0 1. BACKUP je naredba po destruktivnosti slična naredbi FORMAT, pa će, verovatno, biti potrebno da prethodno otkucamo ENABLE da bismo je omogućili. Pošto je zahtevao konačnu dozvolu da napravi kopiju, računar će prepisati kompletan sadržaj diskete 0 na disketu 1 (ukoliko želimo da kopiramo i poleđine ovih diketa, moraćemo da otkucamo i BACKUP 2 4).

Pravljenje kopija diskete je daleko mučnije ako posedujemo samo jednu disk jedinicu. Treba da otkucamo BASKUP 0 0 i računar će naizmenično zahtevati da umećemo disketu koju kopiramo ('source disc') i disketu na koju se kopira ('object disc'). Posle svakog ovog umetanja moramo da pritisnemo neki taster kako bismo računaru stavili do znanja da smo obavili ono što se tražilo od nas. Broj naizmeničnih umetanja direktno zavisi od kapaciteta RAM-a našeg kompjutera i od kapaciteta diska. Ukoliko naš računar ima 48 K memorije od kojih je 40 K slobodno a kapacitet jedne strane diskete iznosi 200 K biće potrebno deset promena.

Tek kada napravimo jednu ili dve kopije sistemske diskete (ukoliko umemo, nije loše napraviti i jednu kopiju na traci), možemo da počnemo eksploataciju nove opreme.

Umesto kasetofona

Svaki vlasnik disk jedinice će novi dodatak najpre koristiti kao ultrabrzi kasetofon. Za to se koriste standardne naredbe SAVE i LOAD iza kojih se obavezno navodi ime programa koji treba snimiti ili učitati. Ukoliko je potrebno, u okviru ovoga imena može da se nađe i specifikacija drajva sa kojim se opšti, kao i direktorijuma u njemu.

Osim naredbi SAVE i LOAD, početnik će brzo upoznati i naredbe DIR i DELETE. Prva od njih prikazuje imena svih programa koji su snimljeni u radnom direktorijumu (ili na čitavoj disketi), dok druga omogućava brisanje nepotrebnih programa — DELETE „PROG" će obrisati PROG. Sa naredbom DELETE treba biti prilično oprezan: ako jednom obrišemo neki program, biće vrlo teško vratiti ga u prethodno stanje. Sam program, primenom naredbe DELETE, naime, ne briše se sam program nego samo podaci o njemu koji su se nalazili u katalogu na početku diskete stradaju. Pomoću posebnih programa ovi podaci bi mogli da se restauriraju, ali je za to neophodno znati lokaciju početka obrisanog programa i njegovu dužinu: osim toga, brljanje po katalogu diskete može da izazove gubitak svih programa na njoj. U svakom slučaju, „PEEK i POKE" naredbe kojima se direktno utiče na sadržaj diskete treba prepustiti

Mini podsetnik

„Komodor 64"

Sporo ali dostižno

Zahvaljujući operativnom sistemu smeštenom u 16 K ROM-a u disku i baferu od 2 K RAM-a, disk jedinica VC 1541 predstavlja inteligentan uređaj. To podrazumeva mogućnost paralelnog rada diska i računara — centralni procesor se može rasteretiti pojedinih U/I instrukcija kao i povezati sa drugim računarima.

Neosporno je da se, radeći sa diskom, prvo nauči snimanje i učitavanje programa, tj. naredbe LOAD i SAVE.

Pomoću džoker znaka * (zvezdica) moguće je „skratiti ime", a sa ? se može zameniti slovo u imenu programa. Program koji se zove „Demonstracija" može se učitati na sledeće načine:

```
LOAD „DEMONSTRACIJA", 8 ili

LOAD „DEMO*", 8
```

Program sa diskete može se učitati na ono mesto sa koga je snimljen sa:

```
LOAD „DEMO*", 8, 1
```

Za naredbu SAVE važe ista pravila kao i za LOAD, samo ovde nema smisla koristiti*. Program se može snimiti sa proizvoljnog mesta u memoriji, na primer:

```
SAVE „DEMONSTRACIJA", 8,1
```

Ukoliko je potrebno snimiti na mesto starog programa, koristi se naredba SAVE @ REPLACE (snimi i zameni), a njen sintaksni oblik je SAVE" @ 0:< IME PROGRAMA >", 8

Međutim, ova olakšica ima jedan neprijatan bag. Na disketama koje su skoro pune SAVE @ REPLACE može da ošteti postojeće programe ili datoteke.

Programi se mogu i verifikovati pomoću naredbe VERIFY koja se zadaje sa:

```
VERIFY"<IME PROGRAMA>", 8
```

Za slanje komande disku neophodno je prethodno otvoriti komandni kanal sa:

```
OPEN <BROJ DATOTEKE>, 8, 15
```

Komanda se prosleđuje naredbom PRINT#. Broj datoteke je proizvoljan i može da varira od 1 do 127.

Za formiranje diskete treba poslati komandu NEW:

```
OPEN 1, 8, 15
PRINT # 1, "NEW: DISKETA-1,00"
```

NEW se može skratiti na N, kao i svaka druga komanda. Proces će biti gotovo kada se ugasi crveno svetlo ili, pak, počne da treperi u slučaju greške.

Ako se formira već korišćena disketa, dovoljno je izostaviti identifikaciju diskete — izvršavanje će biti mnogo brže.

Za brisanje datoteka i programa koristi se komanda

```
SCRATCH: PRINT#1, "SCRATCH <IME DATOTEKE>"
```

Program izbrisan sa SCRATCH može se povratiti uz pomoć posebnog programa.

Pored brisanja, moguće je i promeniti ime datoteci i programu, ili ih kopirati pod drugčijim imenom komandama

```
RENAME i COPY.
PRINT # 1, "RENAME: <NOVO IME=staro ime "
PRINT # 1, "COPY: <NOVA DATOTEKA>=<STARA DATOTEKA>
```

Uzastopna brisanja i snimanja disketa dovode do stvaranja malih džepova od po nekoliko blokova. VALIDATE će reorganizovati njihov raspored na disketi.

```
PRINT#1, „VALIDATE"
```

Komanda INITIALIZE je predviđena da u slučaju poremećaja diska izvrši njegovo ponovno „startovanje".

Posle zadavanja komandi, poželjno je zatvoriti komandni kanal pomoću CLOSE naredbe.

```
CLOSE <BROJ DATOTEKE>
```

Pomoću komandnog kanala moguće je i očitati grešku. Posle nastajanja greške, otkuca se sledeći program:

```
1 OPEN15,8,15
2 INPUT#15,A$, B$, C$, D$
3 PRINTA$, B$, C$, D$
```

A sadrži broj greške (popis grešaka), B sadrži opis greške, C broj staze na kojoj je došlo do greške, a D broj sektora.

Bogdan Petrović

Mini podsetnik

Spektrum

Strujama i kanalima po belom svetu

Sa mikrodrajvom i, uopšte, sa spoljašnjim svetom „spektrum" komunicira preko kanala i tokova podataka (streams). Tokovi podataka se označavaju sa # n, gde je n broj od 1 do 15. Tokove 1, 2 i 3 obično koristi sam bejzik sistem. Znak # je deo same naredbe za OPEN # i CLOSE #.

U proširenom bejziku postoji sedam tipova kanala:

1 tastatura	(k)
2 ekran	(s)
3 ZX štampač	(p)
4 RS232 za tekst	(t)
5 RS232 binarni	(b)
6 mreža	(n)
7 mikrodrajv	(m)

Svaki od navedenih tipova kanala određen je svojim slovom i nije važno da li je otkucano veliko ili malo. Mreža i mikrodrajv zahtevaju dodatne indormacije za kompletnu definiciju kanala.

Kanal za mrežu zahteva i dodeljivanje broja stanice vašem „spektrumu", tako da je sintaksa za njegovu potpunu definiciju „n", x gde je x broj stanice, a kreće se u rasponu 0 do 64.

Kanal za mikrodrajv zahteva da navedemo broj same mikrodrajv jedinice i naziv datoteke, tako da je za potpunu definiciju kanala potrebno pratiti sledeću sintaksu: **„m"; y;** „naziv", gde je y broj mikrodrajv jedinice od **1** do **8,** a „naziv" je slovni niz od 1 do 10 znakova.

CAT y

Daje spisak svih datoteka na kartidžu u mikrodrajv jedinici y. Spisak je sortiran po abecedi i predhodi joj naziv samog kartidža, na kraju je naveden preostali kapacitet kartidža u kilobajtima.

CAT # z; y

Šalje spisak — katalog kartidža u mikrodrajv jedinici **y** na tok (stream) označen brojem **z.**

CLOSE # stream

Raskida vezu bilo kog kanala sa navedenim tokom. Ako postoji bilo kakav sadržaj u baferu, tada se ti podaci ili šalju (na mrežu) ili snimaju (na mikrodrajv).

ERASE „m"; y; „naziv"

Briše datoteku navedenog naziva sa kartidža u mikrodrajvu **y.**

FORMAT „m"; y; „naziv"

Priprema za upotrebu prazan kartidž u mikrodrajvu y. Ime koje se navodi u „naziv"--u kasnije će se pojavljivati u katalogu.

FORMAT „n" ; x

Proglašava dati „spektrum" u okviru mreže jedinicom **x.**

FORMAT „t" ; x
FORMAT „b" ; x

Definiše brzinu prenosa (baud rate) za RS232 interface u skladu sa navedenom vrednošću **x. (X** treba da bude jedna od sledećih standardnih brzina prenosa: 50, 110, 300, 600, 1200, 2400, 4800, 9600, 19200).

INKEY $# stream

Hvata iz niza samo jedan znak, ako takav postoji na datom toku (stream), inače vraća prazan niz " " ako ni jedan znak ne postoji na navedenom toku.

Ova instrukcija ima smisla samo ako je dati tok vezan za mrežu ili RS232 interface.

INPUT # stream ; promenljiva

Uzima navedenu promenljivu sa navedenog toka (stream). Tok prethodno mora biti otvoren i dodeljen nekom od ulaznih kanala.

LOAD * opcije kanala

Učitava program, podatke ili CODE datoteku sa datog kanala. Mogu biti upotrebljeni samo kanali „m", „n", i „b". Sve opcije koje prihvata sintaksa naredbe LOAD prihvata i naredba LOAD*.

MARGE * opcije kanala

Važi sve što je navedeno i za naredbu LOAD*, izuzev što ne briše program ili varijabile koje se nalaze u memoriji, osim što pravi mesto za linije novog programa i njegove varijabile.

MOYE izvor TO odredište

Usmerava tok podataka sa izvora na odredište. Izvor ili odredište mogu biti ili kanali ili tokovi.Dejstvo ove komande prestaje tek kada se na izvoru pojavi oznaka za kraj datoteke ili podataka. Ovo se može desiti samo ako je izvor ili mreža ili mikrodrajv kanal ili tok (stream) povezan sa mikrodrajv kanalom.

Ako su izvor ili odredište kanali, tada se obavlja i njihovo otvaranje za prenos podataka, kao i zatvaranje posle prenosa.

OPEN # stream, kanal

Povezuje navedeni kanal sa datim tokom (stream) da bi omogućio vezu bejzika sa navedenim kanalom radi obavljanja ulaza i izlaza podataka preko tog kanala. Tok mora biti predhodno zatvoren ili otvoren za kanale „k", „s" ili „p".

PRINT # stream . . .

Šalje navedeni niz elemenata PRINT naredbe (. . .) na dati tok (stream). Tok mora predhodno biti otvoren i dodeljen nekom izlaznom kanalu. Niz u PRINT naredbi ima istu sintaksu kao i ranije, a može sadržati još naredbi sa „ " znakom.

SAVE = * opcije kanala

Snima program, podatke ili CODE datoteku na dati nakal. Mogu biti upotrebljeni samo kanali „b", „n" ili „m".

YERIFY * opcije kanala

Isto kao i kod instrukcije LOAD, osim što se podaci ne učitavaju, već samo porede sa podacima u memoriji.

Borivoj Perzić

---

odličnim poznavaocima računara; za početnika je dovoljno da tri puta razmisli pre nego što obriše program!

Osim komande DELETE, vaš DOS verovatno poznaje i naredbe DESTROY ili WIPE. Pomoću ovih naredbi može da se briše veća grupa programa uz korišćenje takozvanih džoker znakova. Džoker znaci (u stranoj literaturi 'wild search characters' — obično zvezdica ili povisilica) zamenjuje bilo koje slovo ili grupu slova u nekom imenu. Naredba DESTROY PR*M će, na primer, obrisati datoteke čija su imena PROGRAM, PROM, PRM, PR12340M i slične. Slično tome, DESTROY A.* će obrisati sve programe koji se nalaze u direktorijumu A, dok bi DESTROY:2.*.*obrisalo sve programe iz svih direktorijuma drajva 2. Naredbe poput DESTROY su, kako im i ime kaže, strahovito destruktivne, pa će računar obično zahtevati od korisnika da još jednom razmisli. Možda će, na primer, biti potrebno da se otkuca ENABLE pre DESTROY, a zatim će biti ispisan spisak programa koji će biti obrisani i biti zahtevano da korisnik pritisne Y (od Yes).

Naredbe poput WIPE su daleko zgodnije: otkucajmo WIPE * i računar će pokušati da obriše sve programe na disketi. Pre svakog brisanja, na ekranu će, međutim, biti ispisano ime programa i zatraženo da korisnik pritisne Y ili N. Ukoliko pritisne Y, program se briše a ukoliko pritisne N — ostaje neizmenjen. Na taj način možemo da obrišemo samo one programe koji nam nisu potrebni.

Da bi se sprečilo nekontrolisano brisanje programa, obično je ostavljena mogućnost da se neki od njih „zaključaju". Kada je neki program zaključan, nije moguće obrisati ga komandama DELETE, DESTROY ili WIPE, niti preko njega snimiti drugi program istog imena — zaključani program je nepromenjiv. Zaključavanje programa nije vrsta njihove zaštite jer postoji naredba za otključavanje programa; zaključavanjem jedino privremeno sprečavamo nehotičnu promenu programa. Da „zaključamo" sve programe obično kucamo nešto poput ACCESS * LOCKED, pri čemu je zvezdica džoker znak koji pokazuje da se naredba ACCESS odnosi na sve programe u radnom direktorijumu. Zaključavanje programa nas neće spasti od nehotičnog formatiranja diskete.

Džoker znaci mogu da se koriste kod mnogih naredbi — ukoliko, na primer, otkucamo LOAD „PR*", računar će učitati prvi program sa diskete čije ime počinje sa Pr; nema nikakve potrebe da kucamo čitavo ime ako prva dva slova jedinstveno određuju program. Kod nekih posebno dobro zamišljenih DOS programa džoker znaci su „inteligentni", naredba RENAME *COMP *RACUN će preimenovati program 1COMP u program 1RACUN, program 2COMP u 2RACUN, a program COMCOMP u COMRACUN!

Dejan Ristanović

Škola sistematskog softvera

Put u središte ROM-a

raspodela memorije

U prvom nastavku naše škole sistemskog softvera smo govorili o organizaciji bejzik interpretatora i operativnog sistema i posvetili posebnu pažnju procesima koji se izvršavaju u trenutku uključenja računara. Rekli smo da operativni sistem ispituje kapacitet priključene RAM memorije i vrši njenu raspodelu. Raspodela memorije je, međutim, daleko složenija nego što smo, govoreći u globalu, u stanju da razumemo. Dovoljan razlog da joj posvetimo posebnu pažnju!

Da bi uopšte bio u stanju da proveri kapacitet priključenog RAM-a, mikroprocesor mora da poseduje minimalan radni memorijski prostor. Za docnije operacije ovaj radni prostor mora da bude daleko veći, pri čemu je njegov kapacitet obično srazmeran mogućnostima računara — jednostavniji računari rezervišu manje RAM-a za rad operativnog sistema, dok će IBM PC utrošiti na njega i dvadesetak kilobajta. Obzirom da veći radni prostor operativnog sistema efektivno smanjuje veličinu programa koji će korisnik razvijati, reklo bi se da treba uložiti mnogo napora u njegovo „sabijanje". No, tako ne mora uvek da bude: korišćenje istih memorijskih ćelija za smeštanje različitih podataka nije u duhu strukturiranog programiranja, što nikako nije rezultat nekih besmislenih standarda; ako je memorijska mapa jasno podeljena na celine, korisniku će biti lakše da menja sadržaje sistemskih promenljivih koje mu pomažu da prestruktuira sistem tako da odgovara njegovim potrebama. Ukoliko se neka sistemska promenljiva koristi za pet raznih stvari u različitim fazama rada računara, korisnik neće moći da je menja bez neželjenih posledica!

Privilegije nulte strane

Pri raspodeli radnog prostora i sistemskih promenljivih treba biti vrlo pažljiv i poznavati karakteristike mikroprocesora. Većina modernih mikroprocesora je, naime, usvojila koncepciju starih „Motorolinih" serija koja je zasnovana na takozvanom 'Zero page' adresiranju. O čemu se radi? Pri radu sa podacima treba utrošiti dva ili čak četiri bajta da bi se navela adresa nekog podatka: instrukcija LDA &2000, na primer, zauzima tri bajta od kojih prvi „čuva" kod instrukcije LDA, a preostala dva adresu &2000. S obzirom da se radni prostor operativnog sistema izuzetno često adresira, bilo bi zgodno da se za ovakve operacije troši manje memorije jer se time nezamislivo mnogo štedi na kapacitetu ROM-a. Zato će mnogi mikroprocesori odvojiti jednu privilegovanu stranu memorije (256 bajta) na koju će se odnositi sve instrukcije u skraćenom formatu. Ta privilegovana strana je obično ona koja nosi broj 0 (adrese &00-&FF) pa će, na primer, instrukcija LDA &20 zauzimati samo dva bajta — prvi sadrži kod instrukcije, a drugi adresu &20. Odakle mikroprocesor može da zna da je &20 adresa na nultoj strani, a ne niži bajt neke „kompletne adrese"? Jednostavno, kod operacije LDA ima u svom sastavu jedan bit koji govori o tome da li je argument smešten na nultoj strani ili nije. Kod mikroprocesora 6502 naredba LDA &2000 će biti kodirana kao AD 00 20, a instrukcija LDA &20 kao A5 20; primetite da &AD možemo da predstavimo kao 10101101 a &A5 kao 10100101.

S obzirom na njenu privilegovanu pozi-

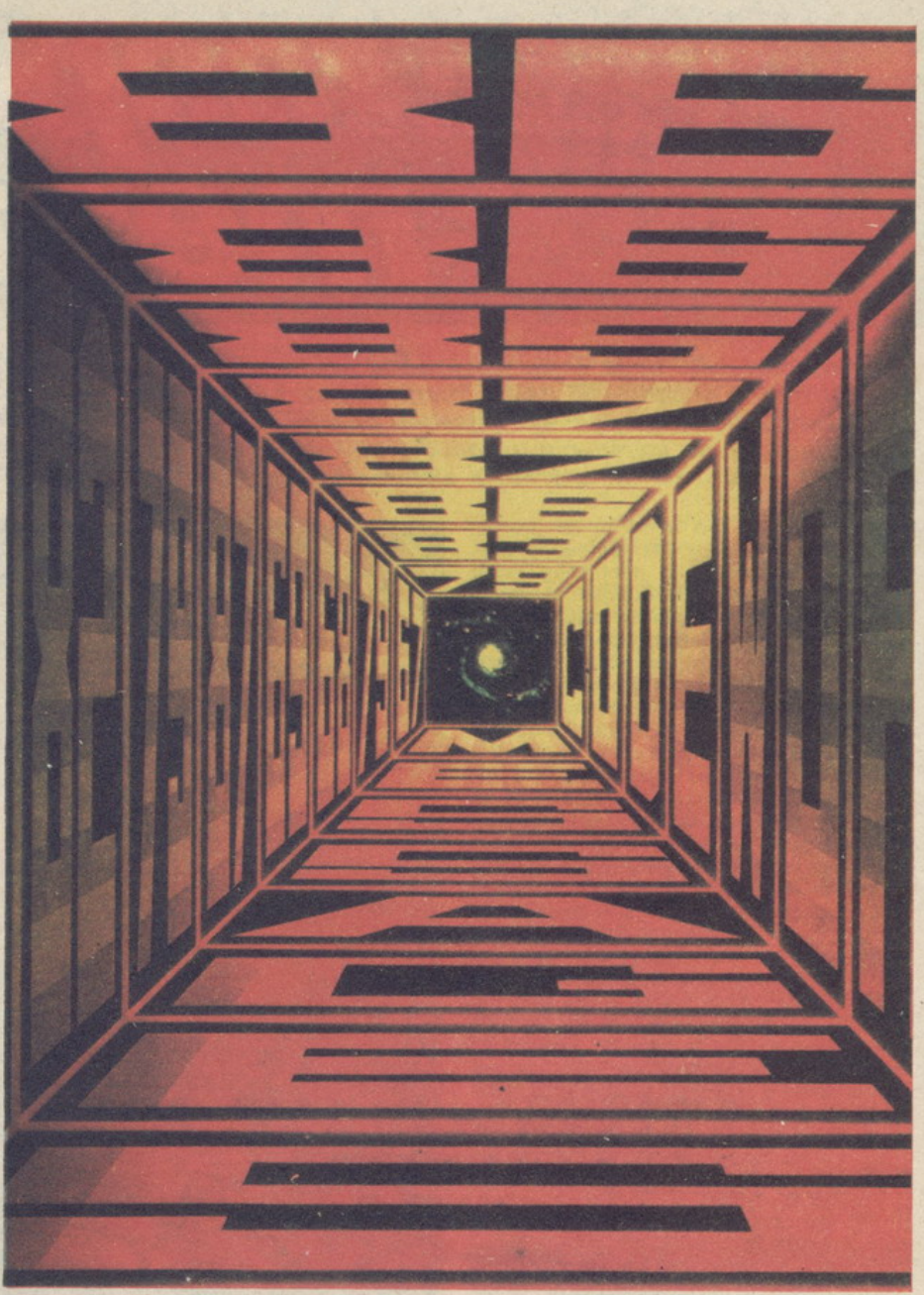
ciju, nultu stranu treba dobro iskoristiti — u njene ćelije ćemo smeštati podatke koji se najčešće čitaju i menjaju. Koji su to podaci? Dobri kandidati za nultu stranu memorije su raznorazni ukazivači: ukoliko formiramo aritmetički stek (mnogo više o njemu docnije), stalno će nam trebati adresa njegovog kraja; ukoliko formiramo stek za instrukcije GOTO i GOSUB, moraćemo da imamo obaveštenje o njegovoj trenutnoj dužini; ako pretendujemo na dobar bejzik, biće nam potrebne tabele imena promenljivih pa i njihove adrese u memoriji . . . Osim ukazivača, na nultoj strani memorije će se naći radni prostor koji koriste razni moduli operativnog sistema kao pomoć pri adresiranju: neki mikroprocesori omogućavaju daleko jednostavnije indirektno adresiranje kada se indirektne adrese nalaze negde na nultoj strani. Dobro struktuirana nulta strana predstavlja, da to ponovimo još jednom, ključ za racionalno napisan ROM.

Rezerva za programere

Iako je memorijski prostor na nultoj strani tesan, ne smemo sebi da dopustimo luksuz da ga potpuno popunimo: bar dvadesetak bajta mora da bude stavljeno na raspolaganje korisniku računara. Kao što mi želimo da sastavimo sistemski softver uz utrošak najmanjeg mogućeg ROM-a, korisnici našeg računara će želeti da pišu programe uz utrošak najmanjeg mogućeg RAM-a, pa će sa zadovoljstvom iskoristiti deo nulte strane koji im je stavljen na raspolaganje. Ukoliko ovaj prostor ne postoji, programer neće imati nikakvog izbora: moraće da koristi neke već upotrebljene memorijske ćelije za rad svog programa, pa će u operativnom sistemu nastati konfuzija i krah od koga će se računar oporavljati jedino posle 'hard reseta' i brisanja kompletne memorije!

Ukoliko želimo da naš računar koristi više ROM-ova koji zauzimaju isti adresni prostor (po koncepciji koja je izložena u prošlim „Računarima"), raspodela memorije na nultoj strani postaje daleko osetljiviji problem. Svaki od ROM-ova sadrži relativno složen program kome će, svakako, biti potrebno mnogo prostora na nultoj strani. „Fer rešenje" je da se pola nulte strane (&80 bajta) odvoji za operativni sistem, a pola za priključene ROM-ove. Obzirom da je bejzik interpretator samo jedan od tih priključenih ROM-ova, operativni sistem ne mora da rezerviše prostor za razne aritmetičke stekove i bejzik promenljive, pa će mu i 128 bajtova verovatno biti dovoljno za normalan rad. U ovom slučaju se obično ne odvaja prostor za korisnikove potrebe, već se smatra da će bejzik interpretator i interpretatori drugih programskih jezika odvojiti deo svojih memorijskih ćelija za korisnika. Tekst procesori i slični u ROM-u smešteni uslužni programi mogu da koriste čitav slobodni deo nulte strane s obzirom da neko ko radi sa njim ne može u isto vreme da piše mašinske programe.

Neki mikroprocesori ne podržavaju adresiranje uz korišćenje nulte strane: popularni Z80 je, na primer, koncipiran tako da je ROM na početku memorijske mape pa su od specijalnog adresiranja ostale jedino RST (ReSTart) instrukcije. Da podsetimo, instrukcija RST je jednaka instrukciji CALL osim u jednoj sitnici: CALL, zajedno sa



adresom, zauzima tri bajta, a RST, opet zajedno sa adresom, samo jedan! Za uzvrat, sa RST se može „skočiti" samo na jednu od osam strogo fiksiranih lokacija u ROM-u, pa ovu naredbu treba čuvati za izuzetno važne i često korišćene potprograme. Pri izboru ovih potprograma treba, ako je ikako moguće, voditi računa i o budućim autorima mašinskih programa: ukoliko očekujemo da će neki potprogram biti veoma često pozivan iz korisničkih programa (kao što je, na primer, „spectrumov" kalkulator), dodelićemo mu neku restart adresu bez obzira na činjenicu da se on ne poziva prečesto iz samoga ROM-a. Ovakva koncepcija će, jasno, biti neostvarljiva ukoliko je kapacitet ROM-a krajnje oskudan ali se takva situacija, s obzirom na cenu EPROM-a, danas javlja relativno retko.

Kuda sa stekom?

Sledeća zona koju treba izdvojiti je mašinski stek. Neki mikroprocesori, poput već pomenutog 6502, za stek rezervišu fiksan memorijski prostor, obično stranu 1. Stek, na taj način, može da ima najviše 256 bajta, što je, zavisno od primene, manje ili više ozbiljno ograničenje. Smeštanjem steka na fiksnu stranu tvorci mikroprocesora štede jedan registar (kod Z80 stek može da se nađe u bilo kojem području memorije, pa je za njega rezervisan ***šesnaestobitni*** registar SP; 6502 za stek rezerviše stranu 1, pa je za pokazivač steka dovoljan ***osmobitni*** registar S), što sa aspekta korisnika nije vredno pažnje. Za programera je, međutim, zanimljivo što fiksiranje steka u neverovatnoj meri smanjuje opasnosti od kraha njegovih programa: ukoliko se u nekoj petlji nađe neka PUSH naredba viška, Z80 će u trenu popuniti stek, a zatim „izbombardovati" i uništiti sve što se nalazi u memoriji. Nasuprot tome, 6502 će popunjavati stek počevši od lokacije &1FF i, bajt po bajt, stići do lokacije &100. Posle sledećeg dekrementovanja, S registar će dobiti vrednost & FF pa će se ponovo popunjavati memorijska ćelija &1FF i tako u krug, sve dok korisnik ne iskoristi taster

povezan sa NM1 nožicom mikroprocesora. Program u memoriji će, posle toga, ostati nedirnut.

Moderni šesnaestobitni mikroprocesori, kao što je Motorola 68000 i National Semiconductors 32016, mogu da smeštaju stek bilo gde u memorijskoj mapi, što je i neophodno kada se uzme u obzir da njihova memorijska mapa obuhvata 16 megabajta lokacija od kojih će, u većini primena, relativno mali procenat biti fizički popunjen RAM-om. Problem „bombardovanja" programa se rešava hardverskim maskama: u toku definisanja radnih uslova mikroprocesora korisnik (ili program u ROM-u umesto njega) proglašava komadić RAM-a ispred (kod većine procesora stek raste prema nižim adresama) steka zabranjenim za upis. Kada se god stek prepuni i mikroprocesor pokuša da upiše nešto u ovo područje, biće generisan takozvani softverski interapt a zatim izvršena odgovarajuća servisna rutina iz ROM-a.

Bilo kako bilo, lokacija steka je tesno povezana sa osobinama mikroprocesora. Određivanje maksimalne moguće veličine steka je, međutim, isključiv problem autora ROM-a i to problem kome treba posvetiti mnogo pažnje. Stek se koristi za mnogo što-šta u ROM-u, pa njegova veličina u mnogome zavisi od organizacije operativnog sistema. Iako ćete u listingu ROM-a nekog računara naići na stotine PUSH i PULL instrukcija, dubina steka koja se na ovaj način iscrpljuje je relativno mala — vrhunac vaših potreba je čuvanje svih registara mikroprocesora na steku, što kod osmobitnih mikroprocesora ne zauzima više od petnaestak bajta. Druga upotreba steka su potprogrami: svaka mašinska CALL instrukcija smešta dva bajta na stek. Stek će se, jasno, dalje puniti ukoliko potprogram pozove sledeći potprogram. Ako ste pristalica strukturiranog programiranja, CALL će biti čest gost u programu koji pišete, pa ćete za stek morati da izdvojite još petnaestak bajtova (možda ćete pomisliti da se neće događati da istovremeno pozovete nekoliko potprograma i da čuvate sve registre, ali je ovakva tvrdnja diskutabilna — moguće je da svaki nivo potprograma poželi da sačuva sve registre!). Obzirom da pri pisanju sistemskih programa ne treba mnogo prepuštati slučaju i da treba biti tolerantan prema budućim potrebama korisnika, reklo bi se da je stotinak (ili 150) bajtova sasvim dovoljan prostor za stek. Postoji, međutim, jedan faktor koji kvari računicu koju smo izveli: interapti.

Gužva na steku

Interapt, kao što mu ime govori, prekida neki posao koji docnije mora da bude nastavljen. Da bi to bilo moguće, rutina koja ga obrađuje mora da sačuva stanje svih registara mikroprocesora koje ima namere da menja i stanje svih flegova. Radi se, dakle, o desetak bajtova koje bismo mogli da upišemo u neku fiksnu memorijsku zonu ali koje uvek smeštamo na stek. Zašto? Ukoliko je kompletno funkcionisanje našeg računara zasnovano na korišćenju prekida (a videli smo da je to koncepcija kako velikih sistema tako i modernih kućnih računara), vrlo je neoprezno zabranjivati interapt u iole dužem vremenskom intervalu. Po prelasku na rutinu za obradu prekida, dalji interapti su automatski onemogućeni. Zamislimo da smo smestili registre u neke fiksne memorijske lokacije a zatim izvršili instrukciju poput EI (Enable Interrupt). Pretpostavimo da uređaj koji je izazvao interapt koji upravo obrađujemo zatraži novi prekid. Jasno je da će novi sadržaji registara zauzeti mesto ranije sačuvanih i da neće postojati nikakav trag o tome da je posao koji se izvršavao pre prvog interapta uopšte postojao. Posmatrajmo, kao suprotnost, situaciju u kojoj se registri (računajući i statusni registar koji obuhvata sve flegove) čuvaju na steku. Po njihovom smeštanju na stek omogućen je, kao i u prethodnom primeru, interapt. Ako novi prekid nastupi pre nego što je tekući obrađen, novi sadržaji registara će biti ponovo bačeni na stek bez gubitka prethodnih. Nema nikakve prepreke da dozvolimo nastupanje daljih prekida. Čim računar bude imao malo mira, završiće obradu poslednjeg prekida i sa steka pokupiti sadržaje registara koji su bili aktuelni kada je pretposlednji prekid prekinut. Prekid po prekid, mikroprocesor će obraditi sve „pozadinske poslove" i preći na dalje izvršavanje glavnog programa.

Znajući da se po nastupanju interapta registri smeštaju na stek, neki mikroprocesori (npr. Motorola 6809) će ovu operaciju izvršiti automatski (čitaj hardverski) što štedi prostor u ROM-u (neznatan dobitak) i ne sasvim beznačajno ubrzava obradu prekida. Po završetku obrade prekida treba mikroprocesoru staviti do znanja da obnovi sadržaje registara, za šta koristimo instrukcije poput RTI (ReTurn from Intrerrupt). Neki drugi mikroprocesori (npr. 6502) će na stek smestiti samo sadržaj statusnog registra, dok ćemo sami morati da čuvamo sadržaj akumulatora i indeks registara ukoliko nameravamo da koristimo neke od njih (u „obilju" registara koje 6502 ima, prilično je neverovatno da ćemo moći da izbegnemo korišćenje makar jednog od njih!). Iz čitave ove priče vidimo da nastupanje prekida teži da velikom brzinom puni stek i to, da stvar bude posebno neprijatna, na nepredvidljiv način.

Nepredvidljivi događaji stvaraju posebne probleme pri dizajniranju ROM-a nekog računara. Sve naše PUSH i PULL instrukcije su prebrojive baš kao i CALL i RET naredbe; kada bi nas neko naterao, mogli bismo da simuliramo razne sekvence i egzaktno utvrdimo potrebnu veličinu steka. Ne možemo, međutim, da predvidimo koliko će brzo korisnik kucati i koliko će, samim tim, biti česti interapti sa tastature. Poseban problem nastaje pri korišćenju nemaskiranih prekida koje mikroprocesor ne može da izbegne instrukcijama poput DI (Disable Interrupt). Komunikacija sa brzim periferisjkim jedinicama kao što su flopi i masivni disk i (posebno) računarske mreže je za mikroprocesor „vanredno stanje" u kome maskirani i nemaskirani interapti nastupaju u nepredvidljivim situacijama i sa nepredvidljivom frekvencijom, pa treba predvideti stek dovoljno veliki da se ni po koju cenu ne prepuni. Ako se stek prepuni za vreme obrade prekida, čak i Motorola 68000 ima dobru šansu da krahira sistem obzirom da će prepunjavanje steka izazivati sostverske interapte koji neće biti pravilno obrađivani. Zbog svega toga, za mašine koje počivaju na interaptima stek od 256 bajta predstavlja sigurnosnu granicu, pri čemu ni daleko veći stek (npr. 1 K) neće nikome smetati ukoliko nismo kritični sa RAM-om.

Proglasivši nultu stranu memorije za radnu zonu i odredivši prostor za stek, rešili smo egzistencijalne potrebe mikroprocesora i možemo da počnemo sa raspodelom memorije za potrebe operativnog sistema. Potrebno je, pre svega, da se pobrinemo za sistemske promenljive, linkove i vektore.

Sistemske promenljive

Nije lako definisati promenljive operativnog sistema tako da se u definiciju uklope operativni sistemi raznih kućnih računara. Posebno je teško odvojiti promenljive operativnog sistema od sistemskih promenljivih koje su ekskluzivno vlasništvo bejzik interpretatora. Bitno je, međutim, shvatiti da se pojam sistemske promenljive bitno razlikuju od pojma radnog prostora: radni prostor se koristi za privremeno odlaganje nekih podataka koji će trenutak docnije biti ponovo pozvani, dok sistemske promenljive čuvaju neke podatke koji se ne menjaju ili se vrlo retko menjaju u toku rada računara.

Postoje, u osnovi, dva razloga za uvođenje sistemskih promenljivih. Neke od njih čuvaju informacije koje je operativni sistem u nekom trenutku generisao uz utrošak značajnog procesorskog vremena i koje će docnije koristiti. Dobar primer ovakve sistemske promenljive je čuveni RAMTOP, memorijska reč u kojoj se čuva adresa poslednjeg memorijskog bajta. Već smo rekli da mikroprocesor po uključivanju računara proverava kapacitet priključenog RAM-a i o njemu obaveštava korisnika. Osim obaveštavanja korisnika, adresa zadnje raspoložive lokacije (ili prve neraspoložive) će biti upisana u RAMTOP. Kada, docnije, bejzik interpretator bude želeo da dimenzioniše neki niz, neće biti neophodno da se ponovo pretražuje memorija — operativni sistem će jednostavno naći razliku sadržaja sistemske promenljive RAMTOP i sistemske promenljive koja čuva podatak o zadnjoj bejzikom zauzetoj memorijskoj ćeliji i, ukoliko je tražena dimenzija veća, prijaviti grešku.

Drugi tip sistemskih promenljivih omogućava korisniku da utiče na rad operativnog sistema. Računar, na primer, može da ima sistemski promenljivu ESCEFC u koju je upisan broj 0 ili 255. Ukoliko je upisan broj 0 taster Escape nema nikakvog dejstva, dok za bilo koju drugu vrednost pritisak na Escape prekida izvršavanje programa. Jasno je da bi kompjuter sasvim lepo funkcionisao i da nema ove promenljive — Escape bi stalno prekidao program. Uvođenje ESCEFC je, međutim, učinilo računar fleksibilnijim utoliko što je korisniku omogućeno da piše programe koji neće moći da budu prekinuti nehotičnim pritiskom na jedan taster uz eventualni gubitak unesenih podataka.

Dok je asortiman i broj sistemskih promenljivih prvog tipa ograničen, sistemskih promenljivih drugog tipa može da bude neverovatno mnogo. Dobar konstruktor ROM-a treba da se potrudi da predvidi što više želja korisnika i uvede sistemske promenljive koje će ih omogućiti (ukoliko, jasno, prostor u ROM-u ili RAM-u nije stešnjen). Ako u nekoj fazi projektovanja sistemskog softvera osetite dilemu 'da li da to uradim ovako ili onako' (a takve dileme ćete osećati gotovo u svakom trenutku rada), nećete pogrešiti ako uvedete jednu novu sistemsku promenljivu; 90% korisnika je neće primetiti ali će preostalih 10% osetiti neverovatno poštovanje prema svo-

Spektrum
interapti u bejziku

me računaru kada je iskoriste i tako reše neki softverski ili, još bolje, hardverski problem. Ono što razlikuje dobro projektovan računar iz srednje ili više klase od lošeg nisu samo memorija i grafička rezolucija, već i fleksibilnost, odnosno prilagođavanje karakteristika operativnog sistema specifičnim potrebama korisnika.

Logička imena

Problem pristupa sistemskim promenljivima je evidentan kod svih kućnih računara. Jasno je da će svaka od njih imati svoje ime (npr. RAMTOP, OSHWM ili ESCEFC) i memorijsku adresu u koju je smeštena. Imena, adrese i funkcije sistemskih promenljivih će biti objavljene u okviru Uputstva za upotrebu pa će korisnik moći da ih čita naredbom PEEK i da menja njihov sadržaj naredbom POKE. Ovakvo pristupanje sistemskim promenljivima neće, naravno, ni malo doprineti boljoj čitljivosti programa: koliko ste puta pokušali da vašem računaru prilagodite neki program pisan za ZX81 i zastali kada ste videli nešto poput POKE 16418,3? Verujete da ova naredba ne radi ništa od posebno opšteg značaja (ona, u stvari, povećava broj linija koje korisnik može da koristi u toku kucanja podataka posle INPUT-a) pa je izostavljate. Program, po prvom startovanju, ne radi; kako da znate da li je to posledica neke greške ili izostavljanja naredbe POKE?

Bilo bi idealno da se u okviru operativnog sistema definišu logička imena svih sistemskih promenljivih, pa bi autor programa, umesto POKE 16418,3, koristio DFSZ=3. Takvo bi rešenje, na žalost, zauzelo dosta prostora u ROM-u (duplo više nego što mislite: svako od imena sistemskih promenljivih mora da figuriše u dve tabele, jednom za slučaj da se nalazi sa leve a drugi put za slučaj da se nalazi sa desne strane znaka jednakosti). Druga mogućnost je da se nekim sistemskim promenljivima dodele logička imena i da se kreira neka opšta naredba koja omogućava pristup ostalima, tako da taj pristup izgleda više struktuirano od naredbe POKE. Kod BBC i Electrona su, na primer, uvedene sistemske promenljive PAGE, TOP, HIMEM, LOMEM i par drugih kojima su dodeljena logička imena, a zatim je kreirana naredba FX koja prosleđuje podatke operativnom sistemu kako bi ovi bili upisani u neku sistemsku promenljivu. Tako je, na primer, *FX 214,3 ekvivalentno sa POKE &266, 10; ova naredba menja sadržaj sistemske promenljive u koju je smeštena dužina tona koji se čuje kada pritisnete CTRL G. Iako ovakvo žongliranje liči na jednostavno preimenovanje naredbe POKE i potvrđivanje onog poznatog programerskog vica o „struktuiranom đubretu", Acornovi konstruktori su imali dobrih razloga za njega: naredba *FX je omogućila kompatibilnost softvera sa 6502 dodatnim procesorom na način u koji se ovde nećemo upuštati.

Već smo rekli da sistemske promenljive mogu odlično da posluže za promenu nekih procedura operativnog sistema; za ove promene se, međutim, mnogo više koriste linkovi i vektori kojima posvećujemo sledeće pogiavlje.

Dejan Ristanović

Rešenje za koje smo se mi opredelili, veoma je slično rešenju Ričarda Tejlora, objavljenom u časopisu „Your Computer", a omogućuje šest novih naredbi: ON ERROR, ON BREAK, AFTER, EVERY, DISABLE i ENABLE.

Neke od ovih naredbi se mogu kombinovati sa standardnim naredbama CONTINUE, STOP, GO SUB i GO TO. Mi ćemo posebno govoriti o svakoj od ovih naredbi, ali pre svega treba da se posvetimo samom programu.

To je, u stvari, niz mašinskih rutina ukupne dužine 1220 bajtova, sa početkom na adresi 64059. Glavni deo predstavlja bejzik interpretator, praktično prepisan iz ROM-a i dopunjen novim naredbama, a jedna posebna rutina (65161), izvršava se pri svakom mašinskom interaptu (procesor je postavljen u interapt mod 2). Grafički simboli u vrhu memorije su sačuvani, a RAMTOP je spušten na 64058.

Formiranje mašinskog koda povereno je bejzik programu čiju listu dajemo. Treba je pažljivo uneti u računar i onda izvršiti RUN. Kroz otprilike jedan minut posle startovanja, pojaviće se poruka „Start tape, then press any key", pod uslovom da ni u jednoj DATA liniji nije otkrivena greška. Posle snimanja, program će od nas očekivati da kasetu premotamo radi verifikacije. Ako sve bude u redu, kasnije ćemo učitavati program sa;

CLEAR 64058; LOAD""CODE

Interapti

Postoje tri vrste interapta koji se ovde mogu kontrolisati. To su:

1. Greške u interpretaciji bejzika
2. BREAK taster
3. Vremenski prekidi

U međuvremenu između interpretacije dve bejzik naredbe, tačnije *neposredno pred izvršenjem* svake naredbe, računar ispituje da li je možda došlo do nekog od navedenih prekida; tj.:

1. Da li je u prethodnoj naredbi bilo grešaka
2. Da li je upravo pritisnut taster BREAK
3. Da li je istekao neki zadati vremenski period

Ako se ništa od svega toga nije desilo, prelazi se na izvršenje predstojeće naredbe. A ako je do interapta došlo, program se obustavlja i preduzimaju se odgovarajuće mere, naznačene naredbama ON ERROR, ON BREAK, itd. Recimo, interapt se prosto može ignorisati, i odmah nastaviti izvršenje programa. Ili se program zaista može prekinuti, sa uobičajenim raportom. A moguće je u interaptu izvršiti neki potprogram, na potpuno isti način kao da je tu umetnuta GO SUB naredba. Po povratku iz takve rutine, nastavak se vrši od naredbe koja je i inače bila na redu da se interpretira. I konačno, umesto GO SUB skoka može se obaviti i GO TO prelaz, čime se bespovratno prelazi na neki drugi deo programa i odatle nastavlja njegovo izvršavanje.

```
 10 REM +----------------------+
 15 REM !    MULTITASKING      !
 20 REM !  BASIC EXTENSION     !
 40 REM +----------------------+
 50 REM
 60 CLEAR 64058
 70 LET a=64059
 80 FOR n=0 TO 19: PRINT n+1
 90 READ k,a$: LET s=LEN a$
100 FOR t=1 TO s-1 STEP 2
110 LET h=CODE a$(t)-48
120 LET l=CODE a$(t+1)-48
130 IF h>9 THEN  LET h=h-39
140 IF l>9 THEN  LET l=l-39
150 LET b=16*h+l
160 POKE a,b
170 LET s=s+b
180 LET a=a+1
190 NEXT t
200 IF s<>k THEN  PRINT "ERROR
in line ";1000+10*n: STOP
210 BEEP 0.2,24: CLS : NEXT n
220 SAVE "multi-task"CODE 64059
,1220
230 PRINT "VERIFY": VERIFY ""CO
DE 64059,1220
1000 DATA 8353,"f32ab25c363e2bf9
2127fce5ed733d5c2101ff1102ff0137
00360023397fedb02100ff223bff223d
ff2240ff2242ff2189fe22fffe3efeed
47ed5efbc3e4fbe7"
1010 DATA 7676,"cdbf16fd340ddf06
00fe0dcaa1fbfe3a28ed3a3a5c3c2858
fe09ca04fc3a3fff2a3bff4404ca04fc
04ca63fb0100ffed433bff22425cfd36
0a00fd3600ff17d2"
1020 DATA 8438,"e4fbfd660de333ed
4b453cc5ed4b7d5cc5ed4b845cc5ed4b
885cc5ed4b575cc50100ffc5e5ed733d
5c21e4fbe5c3021fcd541f381d3a44ff
2a40fffd36001401"
1030 DATA 7690,"00feed4340ff4404
ca04fc04ca63fbc3aefa0608110700dd
2a39ffdd7e02cb47203dcb4f2839ddcb
028ecb572831cb5f2004ddcb0296cb67
c204fcdd6e05dd66"
1040 DATA 10622,"06cd46fbdd2239f
fc3aefadd19e5dde5e1c30139ffed42c
1e1d8dd2101ffc9cd46fb10b7dd2239f
f21e4fbe3df2249ff4fe7fd3600ff79f
efeca61fcfeeaca5c"
1050 DATA 7751,"fcfefac2441bc1cd
321c2a455c226e5c3a475c32705cdffe
```

Iako računari nisu u stanju da istovremeno obavljaju dva posla, ipak je moguće postići efekat multiprogramskog rada, korišćenjem tehnike interapta (prekida). Međutim, kontrola interapta se može vršiti jedino iz mašinskog jezika, tako da je korisnicima bejzika na taj način uskraćeno jedno široko polje mogućnosti. Sve što se od interapta može videti na „spektrumu", jesu uobičajeni prekidi rada prilikom neke greške ili pri pritisku tastera BREAK. Čak nije ostavljena ni mogućnost da se ovakvi prekidi programski analiziraju i rešavaju.

Postoje načini da se ove teškoće potpuno prevazidu. Mašinska rutina koju ovde objavljujemo proširuje „spektrumov" bejzik sa nekoliko novih naredbi za kontrolu interapta.

Kontrola grešaka

Da li je nastupio interapt ili ne, proverava se upravo onim redom koji smo naveli. Prvo se ispituje sadržaj sistemske varijable ERR_NR, čime se testiraju eventualne greške nastale pri interpretaciji prethodne bejzik naredbe. Naredba ON ERROR omogućuje kontrolu ovakvih prekida. Na njenom primeru ćemo ujedno pokazati kako se koriste i sve ostale nove naredbe.

Računar će uopšte razmatrati interapte i prepoznavati nove naredbe samo ako je prethodno izvršeno *RANDOMIZE USR 64059*, čime se procesor postavlja u interapt mod 2, a interpretacija bejzika prenosi u RAM. Međutim, nema nikakvog smisla uraditi to neposredno iz editora, u obliku direktne naredbe, jer će se odmah po izvršenju te linije i povratku radi izdavanja raporta, procesor vratiti u mod 1, a kontrola preneti nazad u ROM. Dobićemo raport „0 OK" i sve će biti kao da ništa nismo ni radili.

Umesto toga, startovanje mašinske rutine treba vršiti u okviru bejzik programa. Na primer, na samom početku:

```
10 RANDOMIZE USR 64059
20 PRINT a
30 STOP
```

Izvršenje ovog programa (sa RUN), daće sasvim očekivan rezultat „2 variable not found", pošto je računar neposredno pred izvršenje naredbe STOP video da je u prethodnoj naredbi (PRINT a) bilo problema, a kako nigde pre toga nismo naznačili *šta* želimo da se u takvim situacijama preduzme, doći će do standardnog prekida.

Ubacićemo zato na liniji 15 naredbu kojom zahtevamo ignorisanje bilo kakve greške u interpretaciji programa. Želimo, dakle, da računar izvrši naredbu STOP bez obzira što je pre toga bilo problema sa PRINT a:

```
15 REM ON EROR: CONTINUE
```

Naredba ON ERROR, a isto važi i za sve ostale, ukucava se slovo po slovo, u okviru *REM* linije. Sasvim je svejedno da li se koriste mala ili velika slova, a i razmak između ON i ERROR nije obavezan. Problema neće biti čak i ako se razmaci ubace potpuno bez reda, ili se na nekim mestima koriste komandni simboli (CHR$ 16 — CHR$ 23). Međutim, prvi važeći karakter iza ON ERROR mora biti „:" (CHR$ 58). Što se tiče naredbe CONTINUE, koja iza toga sledi, to je standardna spektrumova naredba i kao takvu je treba i uneti (kursor će ionako automatski preći u „K", čim se otkucaju dve tačke).

Naravno, u fazi kucanja programa, pod kontrolom ROM-a, može se iza REM naredbe uneti bilo šta. Tek kada računar bude pokušao da to interpretira, greške će biti otkrivene i normalno prijavljene, recimo sa „C Nonsense in BASIC".

Prisustvo karaktera ":" ne treba da zavara. Ne radi se tu o dve odvojene naredbe, već o jednoj jedinoj. Iza REM naredbe *nikada* se ne može dodati još nešto, a da to bude protumačeno kao sledeća naredba u istoj liniji. Međutim, razume se, linija ne mora početi sa REM. Sasvim bi bilo isto i da smo umesto linije 15 dopunili liniju 10:

```
10 RANDOMIZE USR 64059: REM ON ERROR: CONTINUE
```

A može se koristiti i IF naredba, kao recimo:

```
100 IF u$="y" THEN REM ON ERROR: CONTINUE
```

Kako god da se naredba ON ERROR: CONTINUE izvrši, rezultat će biti isti: računar se tako ubedi da ignoriše greške. Međutim, to jedino neće važiti ako je greška otkrivena u poslednjoj naredbi programa, neposredno pred povratak u editor. Recimo, ako se iz našeg primera ukloni linija *30 STOP*, opet ćemo imati raport „2 Variable not found", bez obzira na ON ERROR naredbu. To je zato što se interapti *uopšte ne proveravaju* ako ne predstoji više nijedna bejzik naredba za izvršenje.

Ipak, bilo da je greška bila u poslednjoj naredbi ili ne, i bilo da smo zahtevali ignorisanje grešaka ili ne, desiće se sledeće stvari:

1. Programski kursor se automatski dovodi na liniju sa greškom (to je veoma korisno, jer se tako sumnjiva linija odmah može uzeti pomoću naredbe EDIT, bez dosadne procedure sa naredbom LIST).
2. Linijski broj linije sa greškom smešta

```
cbc28a1cef0238ebcde934d27afa2a555c3ec0a6205bfe01ce0056235eed53455c235e2356eb1923"
1060 DATA 7555,"22555ceb225d5c571e00fd360aff15fd720dca7afa14cd8b19c2ec1bdffe0d28c5fe3aca7afac38aic2a455c226e5c3a475c32705cfdcb0a7e20e12a423ccd6e"
1070 DATA 6713,"197ee6c03a445c26a5f32ab25c363e2bf92b2b223d5c2a6e5c22455c3a705c32475c3e3fed47ed56fbc38313fb2a49ff7ee6effeea28092b110002cd8b191807"
1080 DATA 7746,"2a555c2b225d5c2127fce53a3a5cfe0828973245ff2a455c2246ff22495c3a475c3248ff1883c1e1d17a3cca361fc3261fc1dffe2acaa1fbfe3a2048e7fefec2"
1090 DATA 8789,"8a1ce7fe0dc28a1ce1d17a3c2805d5e5c3e4fbc1ed43575cc1ed43885cc1ed43845cc1ed437d5cd13be3ed733d5ced53425cfd740a2a42ff2240ff2a3dff223b"
1100 DATA 8566,"ffc3e4fb11c7fee5225d5c1a3c28183dca8a1c46cbe8b80813e70828ee1a133c20fb1313e118e0eieb235e2356ebe9113fffcd3ffeed433bffed433dffc3a1fb"
1110 DATA 8065,"1144ffcd3ffeed4340ffed4342ffc3a1fb2a0b5c2323cdb433cd941efe0bd2f924fe08300ccd30fef3dd4e00dd4601fbc92005ed4b46ffc90600fe0920053a48"
1120 DATA 9841,"ff4fc93a45ffc6314ffe3ad8c6074fc9f3cdcefdddcb029efee2200ce7fe0dc28a1cddcb02e61825feed2818feecc28a1cddcb02be180df3cdcefdddcb02defe"
1130 DATA 11003,"edc28a1ccd67fedd7105dd7006ddcb028eddcb02d6fbc3a1fbf3dffe0d280bcdeafdddcb02c6fbc3a1fbcdf1fdcbc61910fbfbc3e4fbf3dffe0d280bcdeafddd"
1140 DATA 8812,"cb0286fbc3a1fbcdf1fdcb861910fbfbc3e4fbcd0ffecd991edd7100dd7001dd7103dd7004ddcb02feddcb02a6e7c9eb2a555c2b18340608210affff110700c9df"
1150 DATA 9521,"ebed4b555ce7ed42d28a1cdffe2cc8fe3a20f2c9cdfafdcd7cfedffe3a3e002814e7cdfafdfe3ac28a1ccd7cfecd941efe08d2f9244f878787e7914f0600dd2101"
1160 DATA 9280,"ffdd09c9dffe3ac28a1ce7eb36ffffed281a34feec28154fe7fe0dc28a1c790100fefee8c804fee2c28a1cc9e7eb2a555c2bcd7cfecd991e210f27ed42daf924"
1170 DATA 7966,"c9a7ed52444dcdb22a051dc3de35fff3dde5e5d5c5f50608dd2101ff110700dd6e00dd66012bdd7500dd74017cb52010ddcb02cedd6e03dd6604dd7502dd7401"
1180 DATA 9672,"dd1910dbf1c1d1e1dde1fbc36f6e6572726f72ffeafc6f6e627265618bffbfc61667465726ff4bfd657665727279ff72fd64697361626c65ff34fd636e61626c65"
1190 DATA 693,"ffb1fd00"
```

se na adresu 65350 (dva bajta), a broj naredbe na 65352 (jedan bajt).

3. Kodirana oznaka greške smešta se na 65349.

Na primer, u našem programu će se programski kursor prebaciti na liniju 20. bez obzira gde se pre toga nalazio. PRINT PEEK 65350+256 PEEK 65351 daće linijski broj 20, PRINT PEEK 65352 daće broj naredbe 1, a PRINT PEEK 65349 daće redni broj greške u tabeli raporta, umanjen za jedinicu.

Postoji i jednostavniji način da se ovi podaci prikupe. Jednostavno treba definisati funkciju:

```
5 DEF FN e(n)=USR 64780
```

Vrednosti funkcije su sledeće:

FN e(8)=linijski broj naredbe sa greškom
FN e(9)=broj naredbe sa greškom
FN e(10)=ASCII kod koji odgovara raportu

Naravno, sasvim je nevažno kojim slovom se funkcija označava. Mi smo koristili „e", da bi nas asociralo na „error".

Naredba ON ERROR: CONTINUE nije, svakako, naročito korisna. Veoma retko smo u situaciji da nas baš ne interesuje do kakve je greške u programu došlo. Ipak, dajemo primer koji tu naredbu uspešno koristi za crtanje grafika nepoznate funkcije. Svaka tačka koja pada van ekrana se i ne uzima u obzir.

```
10 RANDOMIZE USR 64059
20 REM ON ERROR: CONTINUE
30 FOR n=o TO 255
40 PLOT n, 120+80*SIN (n*PI/128)
50 NEXT n
```

Druga naredba, ON ERROR: STOP, vratiće računar u uobičajeni režim rada, kada će svaka greška izazvati prekid programa sa raportom. Na primer:

```
10 RANDOMIZE USR 64059
20 REM ON ERROR: CONTINUE
30 PRINT a
40 REM ON ERROR: STOP
50 PRINT a
60 STOP
```

U ovom programu, samo će linija 30 PRINT a biti ignorisana, dok se *50 PRINT a* neće tolerisati.

Ako nigde u programu ne koristimo ON ERROR naredbu, podrazumeva se ON ERROR: STOP.

Daleko više mogućnosti od pretnodne dve, pruža naredba ON ERROR: GO TO, kao recimo u primeru:

```
10 DEF FN e(n)=USR 64780
20 RANDOMIZE USR 64059
30 REM ON ERROR: GO TO 100
40 INPUT a
50 STOP
1000 REM ON ERROR: GO TO 1000
1010 GO TO FN e(8)
```

Ovaj program se nikako ne može zaustaviti unošenjem STOP u INPUT liniju 40. To je zato što se raport „H STOP in INPUT tretira kao greška i izaziva skok na liniju 1000.

Ono što se iz samog programa ne vidi, a što treba znati, jeste da se neposredno pre skoka izazovnog greškom automatski obavlja ON ERROR: STOP, tako da bi prva sledeća greška dovela do konačnog prekida. Zato na liniji 1000 još jednom stoji ON ERROR naredba, da bi se obnovilo prethodno stanje.

Naredba GO TO FN e(8) na kraju programa izazvaće ponovo skok na liniju 40, i sve će se to ponavljati dok se u varijablu *a* ne unese korektna brojna vrednost.

Ostaje nam još da nešto kažemo o poslednjoj ON ERROR naredbi, a to je ON ERROR: GO SUB. Kao što i treba očekivati, ona dovodi do pozivanja odgovarajuće „interapt rutine" (potprograma), prilikom svake greške. Sledeći program formira slučajne koordinate x i y, a zatim crta odgovarajuću tačku na ekranu. Ako je to nemoguće („B Integer out of range"), izvršiće se potprogram 9000, koji samo štampa poruku o greški, a onda čeka unošenje bilo kog tastera (PAUSE o) da bi izbrisao ekran i vratio se ponovo u prekinut program.

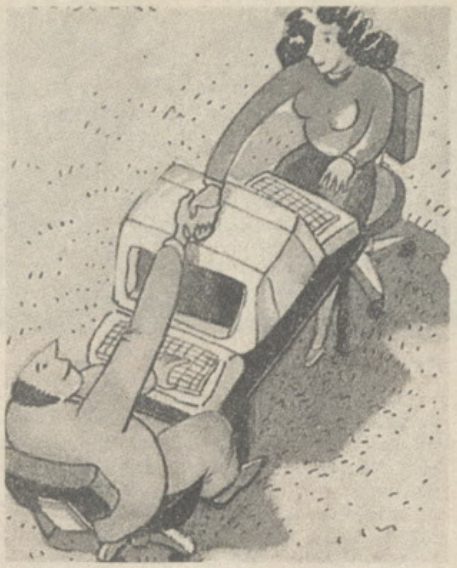
```
  10 DEF FN e(n)=USR 64780
  20 RANDOMIZE USR 64059
  30 REM ON ERROR: GO SUB 9000
  40 LET n=1
  50 PRINT AT 21, 25;n
  60 LET x=RND*300
  70 LET y=RND*200
  80 PLOT x,y
  90 LET n=n=n+1: GO TO 50
9000 PRINT „REPORT "; CHR$ FN e(10);",
";FN e(8); ":";FN e (9): PAUSE O:CLS
9010 REM: RETURN
```

Naredba REM: RETURN koristi se ovde za povratak iz interapta. Kada bi stajalo samo RETURN, imali bismo „7 RETURN without GO SUB", jer zapravo nigde i nismo pozvali nikakav potprogram (skok je usledio zbog greške na liniji 80).

Pre samog skoka, baš kao i u slučaju GO TO naredbe, automatski se obavlja ON ERROR: STOP. To znači da bi svaka greška unutar interapt rutine izazvala prekid sa raportom. Ali zato naredba REM: RETURN *obnavlja* prethodno stanje, bez potrebe da mi intervenišeno. Jedino u slučaju da smo u okviru potprograma izmenili ON ERROR stanje, recimo sa ON ERROR: GO TO 7000, naredba REM: RETURN bi zadržala *to* stanje, a ono pre interapta bi bilo izgubljeno.

Pri pozivu interapt rutine, računar normalno koristi GO SUB stek za smeštanje linijskog broja i broja naredbe na koju će se obaviti povratak. Međutim, pored toga, u stek se smeštaju i vrednosti sistemskih varijabli: COORDS, DF_CC, S_POSN i DATADD, što znači da se za vreme interapta čuvaju vrednosti PLOT i PRINT pozicije, kao i adresa u DATA listi, koju koristi naredba READ. Po povratku iz potprograma, te vrednosti se vraćaju na svoja mesta, tako da se naredbe PRINT, DRAW i READ mogu koristiti nezavisno u glavnom programu i u interapt rutini.

Dodatno korišćenje GO SUB steka samo po sebi isključuje mogućnost povratka pomoću obične RETURN naredbe. Računar prepoznaje podatke u steku i neće dopustiti korišćenje RETURN, ako se radi o interaptu. S druge strane, ako naiđe na REM: RETURN, a u steku nema podataka za povratak iz interapta, već se tamo recimo nalaze samo linijski broj i broj naredbe, spremni za običnu naredbu RETURN, ništa



se neće ni dogoditi. Ne postoji nikakva opasnost da dođe do zabune.

Kontrola BREAK tastera

Naredba ON BREAK ima potpuno istu sintaksu kao i ON ERROR, samo što se odnosi na prekide izazvane pritiskom na BREAK. Tako su moguće kombinacije:

```
ON BREAK: CONTINUE
ON BREAK: STOP
ON BREAK: GO SUB
ON BREAK: GO TO
```

Prva od ove četiri naredbe će izazvati ignorisanje tastera BREAK (što je jedino korisno za one koji vole da zaštićuju svoje programe), druga će vratiti mogućnost prekida programa, a preostale dve će usloviti skok na naznačenu rutinu. Za razliku od naredbe ON ERROR, ovde će se neposredno pre skoka izvršiti *ON BREAK: CONTINUE*, što znači da će nadalje BREK biti ignorisan. Međutim, ako je skok usledio zbog ON BREAK: GO SUB, stanje će biti sačuvano i vraćeno prilikom nastavka programa, tj. prilikom izvršenja REM: RETURN, izuzev ako se u toku potprograma to stanje ne izmeni, u kom slučaju se ono i zadržava (potpuno isto kao i kod ON ERROR).

Ako nigde u programu nije korišćena ON BREAK naredba, podrazumeva se ON BREAK: STOP.

U jednostavnom primeru koji dajemo, program se nikako ne može zaustaviti sve dok ne pritisnemo taster „a". Tek tada se izvršava ON BREAK: GO TO 1000, čime je omogućen prekid:

```
  10 RANDOMIZE USR 64059
  20 REM ON BREAK: CONTINUE
  30 IF INKEY$=„A" THEN REM ON
BREAK: TO TO 1000
  40 GO TO 30
1000 PRINT „break"
```

Treba imati u vidu da naredba ON BREAK kontroliše *samo* prekide programa između dve naredbe, tj. one prekide koji bi inače izazvali raport „L BREAK into program". To ne važi ako se BREAK pritisne u toku izvršavanja naredbi SAVE, LOAD, itd., ili kao odgovor na pitanje „scroll?" (za vreme naredbi PRINT, LIST itd.). Ovakvi slučajevi inače izazivaju raport „D BREAK — CONT repeats" i spadaju pod kontrolu naredbe ON ERROR.

Vremenski interapti

Prekidi koji nastupaju posle isteka određenog vremena, omogućeni su zahvaljujući prisustvu *osam nezavisnih brojača* ili „tajmera". Svaki od tih brojača za sebe broji unazad po jedan, prilikom svakog mašinskog interapta. Da bi se tako izbrojalo do 50, potrebno je, dakle, da nastupi 50 mašinskih interapta, a to znači da protekne jedna sekunda vremena (jer se interapti dešavaju svakih 20 ms). Međutim, ne treba preterano biti siguran u vreme koje protekne između dva interapta. Pojedine bejzik naredbe isključuju mašinske interapte (pomoću mašinske naredbe DI), tako da brojači stoje, a vreme ipak prolazi. Zato ne treba da nas iznenadi ako utvrdimo da spektumov časovnik zaostaje. To unapred i treba očekivati.

Naredbe *AFTER* i *EVERY* mogu raditi sa bilo kojim od osam brojača. Oni su označeni brojevima između 0 i 7, a ako ne navedemo oznaku, podrazumeva se brojač 0. Tako, recimo, naredba AFTER 100: STOP, znači da će program biti zaustavljen čim brojač 0 odbroji sto jedinica. Zapravo, brojač će u početku biti postavljen na vrednost 100, a onda će, pre izvršenja svake bejzik naredbe, kada se i inače testiraju svi interapti, biti provereno da li je brojač stigao do nule dok se izvršavala prethodna naredba. Ako jeste, doći će do interapta.

Sledeći primer koristi naredbu AFTER da bi prekinuo nizanje slučajnih brojeva posle otprilike 10 sekundi od početka, mereno brojačem 7. Oznaka brojača uvek se navodi iza početne vrednosti brojača, odvojena zarezom:

```
10 RANDOMIZE USR 64059
20 REM AFTER 500,7: STOP
30 PRINT INT (RND*100)
40 POKE 23692,255
50 GO TO 30
```

Sam prekid će se u stvari obaviti kao da je prethodno izvršena naredba ujedno bila i poslednja naredba programa, pa sledi raport „0 OK", a ne „9 STOP statement".

Osim AFTER a, b: STOP, moguće su i konstrukcije:

```
AFTER a, b: GO SUB n
AFTER a, b: GO TO n
```

gde je „a" početna vrednost brojača, „b" njegova oznaka (nije obavezna), a „n" linijski broj. Umesto bilo kog od ovih brojeva mogu se koristiti varijable ili čak aritmetički izrazi. To važi ujedno i *za sve naredbe*, a ne samo za AFTER.

Nije moguća kombinacija AFTER a, b: CONTINUE, jer ona i inače nema naročitog smisla: nastaviti program kada brojač „b" odbroji „a" jedinica — to će se svakako dogoditi i bez naredbe AFTER.

Ako se koristi konstrukcija sa GO SUB, onda se povratak iz potprograma vrši kao i ranije, sa REM: RETURN, a ostaće sačuvana PRINT, PLOT i DATA pozcija. Sve interapt rutine, bez obzira kojoj vrsti interapta odgovaraju, *potpuno se isto tretiraju.* Kada se jednom obavi skok, računar doslovno više ne zna zbog čega je skok usledio. Program se nastavlja kao da se ništa nije desilo, a tek ako usledi naredba RETURN ili REM: RETURN, biće proveren sadržaj GO SUB steka i izvršen povratak, ako je to uopšte moguće.

Za razliku od naredbe AFTER, naredba EVERY može samo imati konstrukciju sa GO SUB. Brojač koji se nalazi pod kontrolom naredbe EVERY, takođe će usloviti prekid kada dostigne nulu, ali će odmah *obnoviti svoju vrednost* i nastaviti brojanje iz početka. To znači da će interapti nastupati u pravilnim vremenskim razmacima, što je ujedno i nasličnije samim mašinskim interaptima na spektrumu. Tako se, na primer, može stalno u uglu ekrana imati časovnik, koji će raditi nezavisno od onoga što se dešava u samom programu. Jedino će vreme koje pokazuje takav časovnik biti prilično problematično, pogotovo ako se radi o dužim intervalima. Primer koji mi dajemo još i nije tako loš.

```
  10 RANDOMIZE USR 64059
  20 LET m=0: LET s=0
  30 REM EVERY 50: GO SUB 9000
  40 LET a=0
  50 PRINT AT 21,0;a
  60 LET a=a+1: GO TO 50
9000 LET s=s+1
9010 IF s>59 THEN LET s=0: LET m=m+1
9020 IF m>59 THEN LET m=0
9030 PRINT AT 0,0;"0" AND m<10;m;":";0"
AND s<10; s
9040 REM: RETURN
```

Bez obzira što se u okviru glavnog programa (linije 40—60) obavlja posao apsolutno nevezan za merenje vremena (rezultat toga se vidi u dnu ekrana), u vrhu ekrana je stalno prisutan časovnik koji odbrojava minute i sekunde, koristeći brojač 0.

Pri kreiranju interapt rutine za naredbu EVERY treba voditi računa da izvršenje potprograma ne traje duže od samog intervala između dva interapta, jer se inače nikada neće izvršiti povratak u glavni program. To je zbog toga što se, za razliku od ON ERROR i ON BREAK prekida, ovde, pri skoku na interapt rutinu, vremenski prekidi *ne isključuju.* Ako, na primer, u naš časovnik ubacimo liniju:

```
9035 FOR n=1 TO 200: NEXT n
```

videćemo da će se vreme i dalje lepo odbrojavati, ali „posao" u dnu ekrana stoji. U međuvremenu se svake sekunde GO SUB stek puni podacima za povratak, koji nikada neće biti ostvaren.

Najduži vremenski interval, koji se može ostvariti naredbama AFTER i EVERY, iznosi oko 200 minuta i odgovara maksimalnoj vrednosti šesnaestobitnih brojača (65535). Duži intervali se mogu dobiti brojanjem interapta u bejziku. Recimo, neku varijablu „time" postavimo na nulu, a onda, koristeći naredbu EVERY, svakih 10 minuta uvećavamo tu varijablu za jedan. Kada se dostigne šest, protekao je jedan sat.

Trenutno stanje bilo kog od osam brojača može se uzeti pomoću iste one funkcije e(n), koju smo uveli razmatrajući greške u interpretaciji bejzika. Za vrednosti argumenta *n* između 0 i 7, dobićemo brojno stanje odgovarajućeg brojača u trenutku čitanja.

Naredbe *DISABLE* i *ENABLE* isključuju, odnosno uključuju, pojedine brojače. Striktno govoreći, brojač se nikako ne može zaista „isključiti". Efekat naredbe DISABLE je samo taj da računar prestaje da uzima u obzir stanje naznačenog brojača pri proveri interapta, dok sam brojač i dalje nastavlja da radi. To je sasvim suprotno od mašinske naredbe DI, koja zaista zaustavlja brojanje.

Naredba ENABLE, s druge strane, nalaže računaru da nadalje uzima u obzir stanje navedenog brojača.

Ako se iza DISABLE ili ENABLE ne stavi ništa, podrazumeva se da se naredba odnosi *na sve brojače* istovremeno (a ne na brojač 0, kao što je to bio slučaj sa naredbama AFTER i EVERY). Inače, sve drugo što sledi iza naredbe (osim karaktera ENTER), biće protumačeno kao oznaka brojača.

Ako u prethodni program sa časovnikom ubacimo, recimo, liniju:

```
35 REM DISABLE 0
```

videćemo da do interapta neće dolaziti. Brojač 0 će neprestano brojati po 50 (zbog naredbe EVERY), ali to nikako neće uticati na rad programa (izostaće vremenski prekidi).

Prema tome, naredba DISABLE je podesna ako treba neki blok poštedeti od vremenskih interapta. Na kraju takvog bloka ćemo upotrebiti naredbu ENABLE i tako obnoviti prekide.

Prekidi koji su izazvani stanjima osam brojača, potpuno su nezavisni jedan od drugog. U fazi testiranja interapta (između dve bejzik naredbe), proverava se prvo onaj brojač do koga se stiglo pri prethodnoj proveri. Ako se, na primer, neposredno pre izvršenja neke bejzik naredbe ustanovi da je brojač 5 dostigao nulu, onda će računar upamtiti da je poslednji provereni brojač bio 5, izvršiće se odgovarajući skok, ako je potrebno, a u narednom ciklusu počeće se sa čitanjem brojača 6.

Takođe, interapti uslovljeni greškama ili tasterom BREAK su nezavisni i međusobno i u odnosu na vremenske prekide. Sve ih po volji možemo mešati, a da računaru time ne priredimo nikakve teškoće. Jedino ćemo se mi sami naći u prilično neprijatnoj situaciji ako treba rasplesti sve to i dokučiti zašto se program na nekom mestu zaustavio.

Na kraju bi trebalo reći i to da pri bilo kom zaustavljanju programa i povratku radi izdavanja raporta, pored već pomenutog postavljanja procesora u interapt mod 1, dolazi i do potpunog brisanja mašinskog steka. To znači da korišćenje naredbe CONTINUE u uobičajenom smislu posle prekida neće biti uvek uspešno: izgubljene su sve GO SUB adrese a, osim toga, da bi program bio kompletan, mora se izvršiti ponovo RANDOMIZE USR 64059, što CONTINUE verovatno neće učiniti. Ovo ograničenje zaista nije preterano, pogotovo ako se imaju u vidu ostale prednosti koje program pruža.

Naredba REM može se upotrebljavati i u svom standardnom značenju, za ubacivanje komentara i slično, ali tada prvi znak iza REM (mora biti zvezdica (CHR$ 42). Čim nema zvedice, računar očekuje novu naredbu, pa ako je ne prepozna, prijavljuje grešku.

Jovan Skuljan

na svoju sliku i priliku

Najbolji YU programer 1984.
Majstorije na računaru
Nove naredbe na „spektrumu"

Kako prevariti bejzik

U normalnim uvjetima, kad naiđe na grešku u sintaksi, „spektrum" će to naznačiti flešujućim upitnikom ako je greška nastala prilikom unošenja naredbe, onosno porukom C Nonsense in BASIC ako je greška u programu. Nama je, međutim, cilj da prije prijavljivanja greške natjeramo računar da provjeri ne radi li se slučajno o nekoj našoj naredbi. Da bismo to postigli, bit će najbolje da upoznamo strukturu „spektrumovog" editora: Dok unosimo naredbu, „spectrum" ne kontrolira njenu ispravnost; to čini tek kada stisnemo ENTER. Od tog časa mogu se dešavati različite stvari:

- ako primjeti grešku, ispituje bit 4 sistemske varijable FLAG 32, koji mu govori treba li prijaviti poruku greške ili flešujući upitnik
- ako treba ispisati upitnik, ispisuje ga
- ako treba ispisati poruku, skače na adresu 13Ø3H; na toj adresi nalazi se naredba HALT koja ga prisiljava da sačeka na interapt
- kad interapt stigne, ispisuje se poruka

U slučaju da se greška pojavi pri izvršavanju programa, odmah se skače na 13Ø3H.

Da bismo prevarili računar, treba ga prvo natjerati da i kad greška nastupi pri utipkavanju naredbe skoči na 13Ø3H, a od tamo na našu rutinu za ispitivanje sintakse. To nam omogućava interapt mod 2. U tom modu računar će nakon svakog nailaska interapta skočiti na adresu koju mi odredimo. Na toj se adresi treba nalaziti naša interapt rutina. U njoj prvo treba resetirati bit 4 od FLAGS2, čime navodimo računar da pri svakoj pojavi greške skače na 13Ø3H S obzirom da se tamo nalazi naredba HALT, računar će prvo sačekati interapt, a kad on stigne, skočiti na našu interapt rutinu. Međutim, interapt može nastupiti i u bilo kojem drugom trenutku, pa stoga u našoj rutini treba da provjerimo je li interapt nastupio na adresi 13Ø3H. Ako je, očito je da računar želi prijaviti grešku, pa treba ispitati je li do nje došlo zato što je naišao na neku naredbu. To ćemo postići tako što ćemo utvrditi je li do nje došlo kod nekog karakterističnog znaka za koji smo odlučili da prethodi svim našim naredbama, npr. uskličnika.

Izvršavanje naredbe

Ovisno o tome je li računar naišao na nju za vrijeme izvršavanja programa ili utipkavanja s tastature, skočit ćemo na njeno izvršenje ili nastavljanje linije u programu. U oba slučaja, treba pronaći u našoj tabeli o kojoj se naredbi radi. Tabela je organizirana tako da je svako slovo naredbe predstavljeno velikim slovom, a zadnje slovo je uvećano za 128. Iza njega se nalaze dva bajta koji kazuju na kojoj se adresi izvršava naredba. Na kraju čitave tabele nalazi se broj Ø koji označava kraj. Rutina koja obavlja to traženje po tabeli je SEARCH. Ako ne nađe zadanu naredbu u tabeli, indicirati će to Z (zero) flegom. Inače, rutina ne pravi razliku između velikih i malih slova i ignorira sve razmake u naredbi.

Svaka naredba može imati parametre. Budući da bi bilo dosta komplicirano saopćiti računaru koje parametre treba svaka pojedina naredba, koristit ćemo jednostavniji sistem: uz svaku naredbu može biti zadano do 32 brojevna i 6 znakovnih (string) parametara koji će biti smješteni na stek, a prilikom izvršavanja svaka će naredba s njega uzeti onoliko parametara koliko joj treba.

Nakon utipkavanja naredbe, svaki brojčani parametar treba prevesti u aritmetiku kliznog zareza. To ćemo najlakše izvesti ako pozovemo iz ROM-a DATA rutinu.

Prije izvršavanja neke naše naredbe, treba sve parametre staviti na posebne stekove. Taj posao obavlja rutina PARAMS s pripadnim podrutinama PUSH za brojčane i PH-STR za string parametre. Brojčani parametri se mogu skinuti sa steka rutinom POP a stringovni pomoću POP-ST.

Nakon stavljanja parametra na stekove, može se pristupiti izvršavanju naredbe.

Definiranje naredbi

Program čiji je listing priložen sadrži sve rutine potrebne za rad. On vam, međutim, neće biti od koristi ako ne definirate vlastite naredbe. Kako se to radi, objasnit ću na jednom jednostavnom primjeru.

Pretpostavimo da želimo definirati naredbu koja će obojiti ekran u zadanu boju, ne mijenjajući pri tom njegov sadržaj. Nazovimo tu naredbu PAINT. Prvo trebamo u tebelu naredbi upisati tu naredbu i adresu izvršavanja. S obzirom da je tabela smještena na 60 500 treba (naravno, koristeći neki asembler) napisati:

```
        ORG 60 500
početak tabele
        DEFM „PAIN"
naredba (velikim slovima!)
        DEFB „T"+=80
zadnje slovo uvećano za 128
        DEFW PAINT
adresa izvršavanja
        DEFB #00
kraj tabele
```

Zatim treba na neko slobodno mjesto u memoriji, na primjer 61000, smjestiti samu rutinu PAINT. Za to će nam ponovo trebati asembleri

```
        ORG  60000
START   LD   HL,I_RUPT      Smjesti u memoriju
        LD   (#FDFF), HL    adresu interapt
        LD   A, #FD         rutine i postavi
        LD   I,A            IM 2 tako da sva-
        IM   2              kih 20 ms nastupi
        RET                 skok na I_RUPT
I_RUPT  RES  4,(IY+#30)     Svaka greška mora
                            uzrokovati skok na
                            adresu 1303
        LD   (HL_SAV), HL   Sačuvaj HL
        POP  HL             Uzmi sadržaj steka
        PUSH AF             Sačuvaj AF
        LD   A,H            Ispitaj da li je
        CP   #13            interapt nastupio
        JR   NZ,IRPT_E      na adresi 1303.
        LD   A,L            Ako je, računar
        CP   #04            želi javiti poruku.
        JR   NZ,IRPT_E      U suprotnom iskoči
                            iz interapta
        LD   A,(IY+#00)     Postoji li greška?
        INC  A              Ako ne postoji is-
        JR   Z,IRPT_E       koči iz interapta
        RST  #18            Uzmi adresu na ko-
        DEC  HL             joj je nastala
        LD   A,(HL)         greška. Ako se na
        LD   HL,SEARCH      njoj nalazi us-
        CP   #21            kličnik, skoči in-
        JR   Z,IRPT_E       direktno na SEARCH
        LD   HL,(23647)     Ako je greška nas-
        LD   DE,(23649)     tala izvan E_LINE
        AND  A              ili za vrijeme iz-
        SBC  HL,DE          vršavanja programa
        LD   HL,#1304       onda ispiši poruku
        JR   NC,IRPT_E      greške
        CALL #2530
        JR   NZ,IRPT_E
        LD   HL,#12C3       U protivnom ispiši
                            flešajući upitnik
IRPT_E  POP  AF             Vrati AF i stavi
        PUSH HL             na stek adresu
                            skoka
        LD   HL,(HL_SAV)    Vrati HL, skaniraj
        RST  #38            i skoči na zadanu
        RET                 adresu
SEARCH  LD   DE,60500       Početak tabele na-
                            redbi u DE
        RST  #18            Adresa prije prvog
        DEC  HL             znaka naredbe u HL
        XOR  A              0 u brojač ispravnih znakova
        LD   (COR_CH),A
S_RPTI  LD   (23645),HL     HI u CH_ADD
        LD   C,#00          C je pomoćni brojač točnih znakova
S_REPT  CALL SPC_IG         Zanemari razmake
```

```
        ORG 61ØØØ
PAINT   CALL POP
uzmi parametar
        RET Z
vrati se ako nije zadan
        LD (#58Ø/),ØBC
stavi parametar u prvi atribut
        LD HL, #58Ø/ 0
i prenesi ga na
```

Vjerovatno nema programera kojem nije ni jednom zatrebala bar jedna naredba koju njegov računar ne poznaje. Mnogim komodorovcima je zasmetalo što njihov ljubimac nema baš nikakve naredbe za grafiku, a „spektrumovcima" nedostatak naredbe za brisanje većih blokova programa ili nešto slično. Ponekad se rješenje nalazi u obliku nekakvog „sajmonovog" ili „beta bejzika", no sigurno je mogućnost definiranja vlastitih naredbi najprimamljivija. Svaki programer ima svoja videnja i želi bejzik „na svoju sliku i priliku" Ovaj tekst je namijenjen vlasnicima „spektruma" koji već poznaju mašinski jezik i žele da definiraju vlastite naredbe, ali im nije poznata metoda koja se pri tom koristi. U njemu je, u najgrubljim crtama, opisana tehnika primjenjena u radu koji je na nedavno završenom konkursu „Galaksije" za najbolji YU program osvojio prvu nagradu.

```
        CALL  CAPS      Pretvori malo slo-
                        vo u veliko
        EX    DE,HL     Tabela u HL, adre-
                        sa znaka u DE
        LD    B,(HL)    Uzmi znak iz tabele
        RES   7,B       i zanemari indika-
                        tor kraja naredbe
        CP    B         Usporedi ga s zadanim znakom
        JR    NZ,NEXT   Ako su različiti,
                        traži slijedeću
                        naredbu u tabeli
        INC   C         Uvećaj brojač toč-
                        nih znakova
        BIT   7,(HL)    Zadnji znak?
        JR    Z,SRC_NX  Ako nije, pređi na
                        slijedeći
        LD    A,(COR_CH) Da li se nađena
                        naredba najbolje
        CP    C         slaže sa zadanom?
        JR    NC,NEXT   Ako ne, traži drugu
        ID    a,c       Spremi brojač toč-
        ID    (cor_ch),a nih znakova i ad-
        LD    (COR_CH),A resu zadnjeg znaka
                        zadane naredbe
        INC   HL        Uzmi adresu izvr-
        LD    E,(HL)    šavanja naredbe i
        INC   HL        spremi ju u JUMP
        LD    D,(HL)    i pređi na slije-
        LD    (JUMP),DE deću naredbu u
        JR    NEXT_1    tabeli
SRC_NX  EX    DE,HL     Zamijeni HL i DE
        INC   DE        i pređi na slije-
        JR    S_REPT    deći znak
NEXT    XOR   A         Da li si stigao do
        CP    (HL)      kraja tabele? Ako
        JR    Z,FOUND   jesi, završi
NEXT_2  BIT   7,(HL)    Traži početak sli-
        INC   HL        jedeće naredbe u
        JR    Z,NEXT_2  tabeli
        INC   HL
NEXT_1  INC   HL
        EX    DE,HL     Zamijeni DE i HL
        POP   HL        Uzmi adresu zadane
                        naredbe
        JR    S_RPT     Nastavi traženje
FOUND   POP   HL        Isprazni stek
        LD    HL,(CHAR_A) Adresu zadnjeg
        LD    (23645),HL znaka zadane nared-
                        be stavi u CH_ADD
        LD    A,COR_CH) Jesi li prepoznao
        OR    A         zadanu naredbu?
        JP    Z,#1304   Ako nisi, javi
                        grešku C
        LD    (IY+#00)#FF Indiciraj „nema
                        greške"
        CALL  #2530     Ako izvodiš prog-
        JR    NZ,RUN    ram, izvrši naredbu
        LD    HL#FFFE   Indiciraj „direkt-
        LD    (23621),HL na naredba"
        CALL  SPC_IG    Zanemari razmake
        CP    #0D       Kraj linije?
        JP    Z,#12CF   Izađi ako je
        LD    HL,#1287  Postavi stek kao
        PUSH  HL        što je normalno
        CP    #3A       Kraj naredbe?
        JP    Z,#1B29   Izađi ako je
        RST   #18       U protivnom, izađi
        DEC   HL        kroz DATA rutinu
        LD    (23645),HL da pokupiš sve pa-
        LD    A,#E4     rametre zadane uz
        JP    #1B33     naredbu
RUN     LD    HL#1303   Postavi stek kao
        PUSH  HL        što je uobičajeno
        LD    HL#1B32   prilikom izvršava-
        PUSH  HL        nja programa
PARAMS  LA    HL#FF00   Stek za brojeve
        LD    (STACK),HL stavi na FF00
        LD    L#40      a stek za stringo-
        LD    (STR_ST),HL ve na adresu FF40
NX_PRM  CALL  SPC_IG    Zanemari razmake
        CP    #0D       Kraj linije?
        JR    Z,RUN_1   Izađi ako je
        CP    #3A       Kraj naredbe?
        JR    Z,RUN_1   Izađi ako je
        CALL  #24FB     Pozovi rutinu iz
                        ROM-a za uzimanje
                        parametra
        BIT   6,(IV+#01) Ako je parametar
        JR    Z,STRING  string, izvrši skok
        CALL  #2DA2     U suptornom, pret-
        CALL  PUSH      vori ga u intidžer
                        i stavi na stek
P_CONT  RST   #18       Ako ima još zada-
        CP    #2C       nih parametara,
        JR    Z,NX_PRM  skoči natrag
        JR    RUN_1     U suprotnom izvrši
                        zadanu naredbu
STRING  CALL  #2BF1     Početnu adresu i
        CALL  HP_STR    duljinu stringa
                        stavi na stek
        JR    P_CONT    Skoči natrag
RUN_1   LD    HL,(JUMP) Uzmi adresu izvr-
        JP    (HL)      šavanja naredbe i
                        skoči na nju
PUSH    LD    HL,(STACK) Adresa steka u HL
        LD    A,L       Ako više nema mje-
        CP    #40       na steku, javi
        JR    Z,ERR_4   grešku 4
        LD    (HL),C    Stavi broj iz BC
        INC   HL        registarskog para
        LD    (HL),B    na stek i uvećaj
        INC   HL        HL za dva
        LD    (STACK),HL To je nova adresa
        RET             steka
POP     LD    HL,(STACK) Adresa steka u Hl
        XOR   A
        OR    L         Ako je stek prazan
        RET   Z         vrati se
        DEC   HI
        LD    B,(HL)    Broj sa steka sta-
        DEC   HL        vi u BC i smanji
        LD    C,(HL)    HL za dva
        LD    (STACK),HL To je nova adresa
        RET             steka
PH_STR  LD    HL,(STR_ST) Adresa steka u HL
        LD    A,#58     Ako je stek popu-
        CP    L         njen, javi
        JR    Z,ERR_4   grešku 4
        LD    (HL),E    Na stek prvo sta-
        INC   HL        vi početnu adresu
        LD    (HL),D    stringa (iz DE)
        INC   HL
        LD    (HL),C    A zatim i njegovu
        INC   HL        duljinu koja se
        LD    (HL),B    nalazi u BC
        INC   HL        Nova adresa steka
        LD    (STR_ST),HL je za 4 veća od
        RET             STARE
POP_ST  LD    HL,(STR_ST) Adresa steka u HL
        LD    A,#40
        CP    L         Ako je stek prazan
        RET   Z         vrati se
        DEC   HL
        LD    B,(HL)    Duljina stringa
        DEC   HL        ide u BC
        LD    C,(HL)
        DEC   HL
        LD    D,(HL)    a početna adresa
        DEC   HL        u DE
        LD    E,(HL)    Na kraju, nova
        LD    (STR_ST),HL adresa steka je za
        RET             4 manja od stare
ERR_4   RST   #08
        DEFB  #03
SPC_IG  RST   #20       Je li slijedeći
        CP    #20       znak razmak? Ako
        JR    Z,SPC_IG  je, traži dalje
        RET             Ako nije, vrati se
CAPS    SP    #61       Ako je znak malo
        RET   C         slovo, pretvori ga
        CP    #7B       u veliko
        RET   NC        U suprotnom, osta-
        AND   #DF       vi ga nepromije-
        RET             njeno
```

LD DE, #5801
slijedećih 767
LD BC, #2FF
bajtova. Time je

LDIR
ekran obojen
RET

Korišćenje programa

Poslije asembliranja, program aktiviramo s RANDOMIZE USR 60000 Nakon toga možemo kotistiti sve naredbe koje smo sami definirali ravnopravno s normalnim nared-

```
COR_CH DEFS 1
HL_SAV DEFS 2
CHAR_A DEFS 2
JUMP   DEFS 2
STACK  DEFS 2
STR_ST DEFS 2
```

Rezerviraj memoriju za neophodne nove sistemske varijable

bama bejzika. Važno je zapamtiti da se ispred svake naše naredbe mora nalaziti uskličnik. Iza same naredbe navodimo sve potrebne parametre, znakovne i brojčane, jednostavno ih odvajajući zarezom.

Kao na primjer, uzimamo već spomenutu naredbu PAINT. Ako otipkamo:

!PAINT 5

svi atributi poprimit će vrijednost 5, što znači da će ekran biti crn sa cijan slovima. Budući da program ne pravi razliku između velikih i malih slova i ignorira razmake, mogli smo isto tako napisati:

!P al nt 5

Treba napomenuti da se razmak ne smije nalaziti odmah iza uskličnika.

Ukoliko zadamo više parametara nego što je potrebno računar će uzeti one zadnje. Tako će

!PAINT 1, 2, 3

imati isti efekat kao i

!PAINT 3

Uzrok ovoj prividnoj nelogičnosti je u LIFO sistemu (Last In First Out) koji je korišten. To, u stvari, znači da će zadnji parametar stavljen na stek biti prvi skinut s njega. Iz iskustva znam da je to najbolji način rada sa stekom.

Nadam se da vas je ovaj članak i program bar donekle uputio u mogućnost definiranja vlastitih naredbi. Isti princip koristi i Hyper BASIC, prvonagrađeni program na „Galaksijinom" konkursu. Jedina je razlika što Hyper BASIC ima već definirane naredbe koje omogućuju definiranje novih i bez poznavanja mašinskog jezika.

Igor Fischer

NOVO! NOVO! NOVO! NOVO! NOVO! NOVO!

KOMPJUTERSKA POČETNICA

autori:
Valentin Čip
Tajib Šahinpašić

Knjiga namjenjena svima vama koji želite znati šta je kućni računar, kako radi, šta je upis podataka, čitanje memorije, osnovi programiranja BASIC instrukcije, riječnik informatičarskih termina.....

Iz ove knjige možete učiti VI i vaše dijete!

NARUDŽBENICA

Tajib Šahinpašić, 71000 Sarajevo, ul. Ivice Marulića 5/3

Neopozivo naručujem knjigu „Kompjuterska početnica" po cijeni od 680,- din. koju ću platiti poštaru prilikom prijema — pouzećem.

Ime i prezime_____

Pošt. br. i mjesto_____

Ulica i broj_____

Potpis i br. l.k._____

hajde da se igramo

Spektrum

BEACH HEAD

Pravac plaža

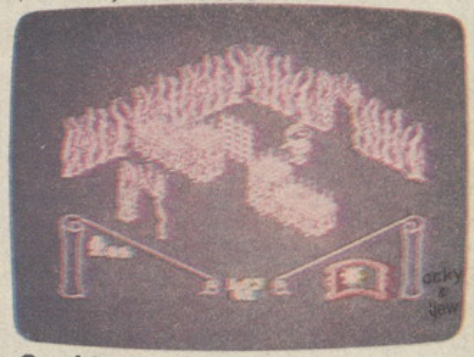
O, kako to neodoljivo podseća na iskrcavanje u Normandiji! Da biste mogli da obavite samo iskrcavanje, potrebno je, jasno, da flotu najpre dovedete, što manje oštećenu, do obale. Da sve to ne bi izgledalo kao najobičnija turistička plovidba, pobrinuće se dobro naoružan neprijatelj, koji će vam pružiti priliku da vodite borbe u svim mogućim uslovima. Prvo vas očekuje prolaz kroz dosta uzan kanal pun neprijateljskih brodova koje ne smete dodirnuti (pecizno rečeno, vi ih smete dodirnuti, ali se tada možete oprostiti sa jednim od svojih života). Uspeli ste da prođete? Divno, ali nova nevolja je već na vidiku — bombarderi koji u niskom letu ispuštaju bombe na vas. Posedujete dosta efikasno oružje, a od vas se zahteva „jedino" preciznost pravog artiljerca, jer avioni se iznenada pojavljuju nad horizontom, da bi vam već za nekoliko trenutaka bili iznad glave i van domašaja. Kako je prikaz trodimenzionalan, potrebno je pogoditi ne samo visinu na kojoj se avion nalazi, već i daljinu, kontrolišući i menjajući nagib topovske cevi. Nakon što uspete da oborite određen broj aviona, od vas se očekuje da isto toliko uspešno preuzmete kontrolu nad tenkovima i, probijajući se kroz minska polja i slične prepreke, dođete do krajnjeg cilja, bunkera negde na plaži koga treba uništiti. Prebrodite li i ovu prepreku, pogledajte rezultat i pomislite kako „verovatno može da se dobije još više poena", promenite težinu igre na „veoma teško", ili, siti ovakvih i sličnih „muških — pucačkih igara", isključite računar i pokušajte da napišete sopstveni program, pa makar se on sastojao i samo u ispisivanju vašeg imena desetak puta.

Spektrum

KNIGHT LORE

Viteška učenost

Još jedan istinski biser iz niske softverske kuće „Ultimate". Zadnja rečenica iz programa „Underworld" (vidimo se u „Knight Loreu"!) se, evo, ostvarila i zaista je teško za ovu igru naći prave reči hvale. Igru započinjete na jednom od četiri startna mesta u zamku koji broji ni manje ni više nego 130 soba! Na početku igre ste naoružani samo istraživačkom upornošću da u tako nepoznatim i mističnim uslovima dođete do željenog cilja, a on je, moramo priznati, u tom trenutku daleko od vas desetinama svetlosnih godina. Cilj je da se, krećući se ka sredini zamka (treba se tu orijentisati među silnim sobama), kao prostoriji gde se nalazi čarobnjak, sakupite 14 predmeta (dijamant, cipela, čarobna kugla . . .) i tako se oslobodite prokletstva, koje se nadvilo nad vas po ulasku u zamak. Prva od smetnji je pravilo da odjednom možete nositi najviše 3 predmeta, što stravično komplikuje stvar, pa će vam najverovatnije (za ozbiljno igranje) biti potrebni papir, olovka i mapa zamka. Jasno, ovde su preskočene razne muke koje vas očekuju kako u sobama gde se nalaze predmeti tako i u svim ostalim prostorijama. Radi se gvozdenim duladima koja preteći padaju sa tavanice, duhovima, sablasnim vojnicima-čuvarima, bodljama koje krajnje nezgodno vire iz zemlje i sl. U ponoć se pretvarate u vukodlaka i tek sa jutrom dobijete svoje normalno, simpatično obličje. Istaknimo i zaista virtuozno, krajnje pedantno urađenu grafiku, kao i veoma uspeli trodimenzionalni način prikazivanja. Vidimo se u . . .

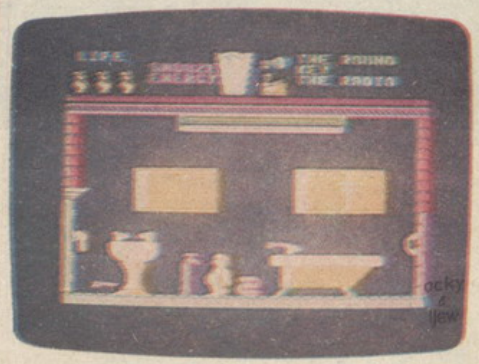


Spektrum

PYJAMARAMA

Pidžamarama

Rano jutro, ali u vašoj kući ne vlada mir. Puna je najraznovrsnijih članova brojne ekipe za ometanje: četkica za zube, duhova, biljki žderalica, pečenih pilića. Njihov cilj je da vas ometaju (šta se drugo može očekivati od jedne ekipe za ometanje), a vaš da probudite svog gazdu Vilija da bi mogao na vreme da stigne na posao. Dok ne uspete da otkrijete sve tajne ove igre, vaš gazda će sigurno dobiti popriličan broj otkaza. Soba ima puno, ali to nije sve. Zamak vašeg gazde, pored silne gomile soba, ima i hangar za poletanje rakete(!?) Predmeti koje sakupljate (što nije baš uvek najjednostavnije) donose vam određene privilegije — ključ, na primer, omogućava prolaz kroz horizontalna vrata . . . Očigledno vam se žuri, jer ste sve vreme u pidžami. Ako i pored sve panike da probudite gazdu dok nije kasno mislite da imate vremena i za igru, nema problema: potrebno je da samo iz jedne od soba nabavite džojstik, dođete do prostorije „video



room" i tu odigrate neku skraćenu verziju „osvajača". Nemojte se osećati previše sigurno čak ni dok hodate na izgled mirnim sobama — vrlo se lako može desiti da iz zemlje (tj. iz poda) izviri ruka i ščepa vas za nogu. Grafika i animacija su odlični, a ekrani zanimljivi, urađeni u puno boja i sa obiljem dopadljivih detalja(na primer, paučina po zidovima).

Spektrum

SKOOL DAZE

Od kolevke pa do groba . . .

Hmm! Ako je verovati mom znanju engleskog jezika — verovatno ćete pomisliti vi — ovo bi trebalo da se piše „school days", ali cela igra je jedna parodija na tradicionalno englesko školstvo, pa je i naslov u tom stilu. Pre početka igre, imena ličnosti se mogu promeniti, što joj daje posebnu draž. Držimo li se originala, vi ste Erik nevaljac, čije je užasno svedočanstvo zaključano u direktorovom sefu. Vaš je zadatak da ga izvučete odatle i uništite. Samo nastavnici znaju šifru sefa i, da biste je otkrili, morate praćkom poskidati sve zastavice po školi. Poseban slučaj je nastavnik istorije Krik, koji je zaboravan, pa ga morate naterati da napiše svoj datum rođenja da bi se setio svog dela šifre. U međuvremenu morate krajnje savesno da prisustvujete školskim časovima sedeći mirno na svom mestu i odgovarajući na pitanja koja postavlja nastavnik. Za svaki prekršaj koji načinite sledi (staromodna) kazna — morate prepisati određeni broj redova. A prekršaji mogu biti zaista različiti — od bežanja sa časa, tuče sa drugovima, ulaska u privatne prostorije nastavnika, do gađanja profesora praćkom. Kada broj redova pređe (impozantnu) cifru od 10 000, uz pesmu „Zbog školo nisam te ni vol'o" bivate izbačeni iz škole. Sama škola, data je u preseku na tri ekrana i na tri sprata, gde su smeštene sve te (m)učionice. Kako to lepo piše u jednoj engleskoj reklami, dan kada ste nabavili „Skool daze" verovatno neće biti najsrećniji u vašem životu, ali će vam sigurno pružiti mnoge prijatne i zabavne sate. A onda, nakon igre, zagrejte stolicu i knjigu u šake!

Spektrum

Moon alert

Uzbuna na Mesecu

Tipičan svemirski ambijent, ali dovoljno blizak — reč je o površini Meseca. Uzbuna! Od vas se zahteva da se u što kraćem vremenu spojite sa matičnom bazom. Kako na Mesecu nema autoputeva (bar ne zasad — dok je ovo izdanje u štampi), opremljeni ste za ovakve prilike adekvatnim vozilom — lunohodom. Njime možete preleteti neki manji krater ili se, zbog loše procene, razbiti o mesečevo stenje. Pored kratera, na putu će vam se naći i sitno kamenje, kao i stene koje je neophodno dematerijalizovati ubojitim laserom. Vaša ruta je podeljena na međustanice — kada god dođete do neke od tih stanica, dalji nastavak igre, ukoliko izgubite život, započeće od zadnje osvojene stanice. Tako se od igrača zahteva pamćenje gotovo cele staze (odlično ukomponovan element u akcionu igru), otprilike po principu „posle baze c moram da povećam brzinu na maksimum i preskočim šiljak, a zatim da usporim i pucam u stenu, uskočim u krater i sl." Reč-dve i o posetiocima zbog kojih je i nastala uzbuna. Na izvesnim deonicama biće vam verni i krajnje neugodni pratioci, tako da ćete, pored sve muke i pažnje usredsređene na „neravan put", morati da se osvrćete i na njih. Oni će pucati na vas, a znaju i da bacaju bombe koje će, tik ispred vas, stvarati male kratere. Po tom pitanju nema diskusije, jer ovde, očigledno, važi deviza „samo uništen UFO je dobar UFO (šta li na ovo kažu borci za miroljubive međuplanetarne odnose?).

Spektrum

FRED

Fred

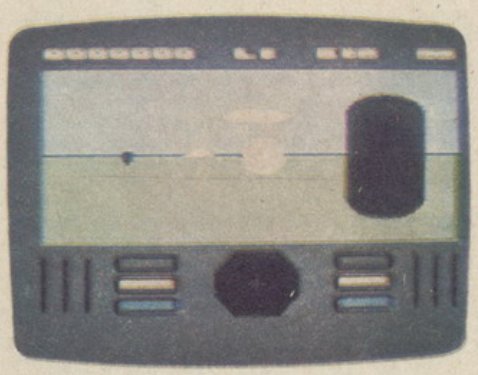
Ovoga puta se nalazite u koži malog simpatičnog istraživača (možda ne baš tipa Indijane Džonsa, ali . . .) koji se odaziva na ime Fred. Recimo da kao svaki pravi istraživač čuvate tačnu lokaciju svog mesta kao tajnu, ali može se naslutiti da je u pitanju Egipat i gomila podzemnih hodnika neke piramide. Nepogrešiv njuh vam govori da tu ima zlata i raznih basnoslovno urednih figurica. Ne varate se, i to, u stvari, i jeste vaš cilj. Blago čuvaju duhovi, neke podzemne životinjice i — kapi (!?). Ne znamo kakva je hemijska priroda ovih kapi koje padaju u pojedinim delovima hodnika, ali su očigledno opasne jer vam smanjuju životnu energiju, što se događa i kada dozvolite sebi bliski kontakt sa duhom. I dok se mukotrpno penjete uz kanape, mimoilazeći zidove i ulazeći u druge hodnike, duhovi se mogu iznenada pojaviti, jer za njih zidovi ne predstavljaju nikakav problem (uostalom, nisu bili nikakav problem od kad duhovi i postoje). Pištolj koji tako odlučno držite ispred sebe možete upotrebiti samo šest puta, jer toliko imate municije. U potrazi za plenom, možete naći i mapu hodnika, što vam znatno olakšava dalje traganje. Nakon gubitka celokupne energije „koja život znači", računar će vam, uz poruku „game over", odsvirati posmrtni marš — hm, nimalo veselo, ali neka — pokazaćete vi njemu već u sledećoj igri! Grafika je vrlo dobra sa zamerkom koja se odnosi na pomeranje ekrana.

Komodor 64

ENCOUNTER

Susret

Praznik za ljubitelje 3-D igara! Podmažite svoje džojstike i dovedite svoju koncentraciju na najviši nivo — predstoji vam uzbudljiv susret sa došljacima iz hipersvemira. Izvanredna 3-D animacija pruža izuzetan osećaj kretanja u svim pravcima (čak i u rikverc!) i svim brzinama (od puže do kosodižuće). Polje za bitku nije ravno, već je „zasađeno" stubovima. Precenivši svoje mogućnosti, prevelikom brzinom ulećete pravo u jedan od stubova i . . . ništa se ne dešava! Pa da, oni samo služe za stvaranje 3-D efekta, samouvereno zaključujete. A sada se malo odmaknite i pucajte u jedan od stubova svojim razornim bombama i gle — one će se od stuba odbijati poput teniskih loptica, nastavljajući put odbijanjem od ostalih stubova (ukoliko su to uopšte normalni stubovi, a, po svemu sudeći, nisu), pa će, najverovatnije, veoma brzo postati opasnost i za vas same. Više nepreciznih hitaca — manje šanse da se preživi. Dakle, ovde „ne pali" rapidna paljba. Čuvajte se i svemiraca kamikaza (jedno oko vam mora neprekidno „visiti" na radaru). Uspete li da ispunite zadatak, tj. da uništite određen broj svemiraca (10—17), na vreme se pripremite za prolazak kroz tunel koji vodi kroz svemir na drugi nivo. Od vas se još jedino (!?) zahteva da uz stravičnu brzinu koju diktira kompjuter prođete čitavi kroz polje puno nekakvih kugli o koje ne smete da se okrznete. Nije da je nemoguće, ali . . . recimo onda bar da je na granici mogućeg.



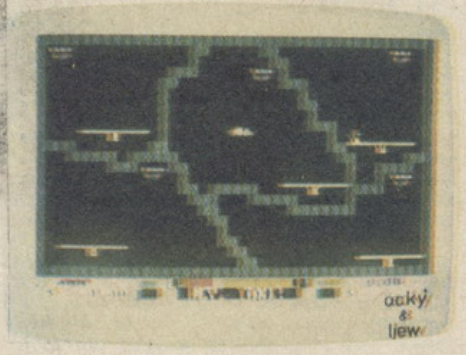
Komodor 64

SPACE TAXI

Svemirski taksi

Kako vam se čini ideja da se oprobate kao vozač taksija? Doduše, ne baš pod uobičajenim uslovima — vaš taksi ima raketne motore i vozi samo zvezdanim stazama. Kada se nađete na jednom od 25 različitih nivoa, dobro osmotrite raspored platformi, jer ćete već sledećeg trenutka čuti poziv „Taksi, molim!". Da, nije greška, baš ćete ga čuti — zahvaljujući dobroj sintezi glasa. Nakon toga se treba što hitnije, ali istovremeno i što mekanije, spustiti na platformu gde se nalazi vaš putnik, koji je, u međuvremenu, počeo nervozno da lupka nogom. Očigledno vam brzina nije neka jača strana. Uostalom, od vaše efikasnosti zavisi i vaša zarada (u poenima, naravno). Sada će vas on zamoliti da ga odvezete na neku drugu platformu, koja zna da bude „jezivo" smeštena na ekranu. U vezi sa sintezom glasa, treba još spomenuti i ono „hvala!" kojim ćete biti nagrađeni za uspelu vožnju (nije sve u novcu!) i uzvik koji otprilike zvuči „aaaughh!" kada, krajnje nemarno, umesto na platformu, aterirate svojoj mušteriji na glavnu. Otežavajućih okolnosti ima zaista puno (platforme koje se pomeraju, krajnje uzani prolazi . . .), tako da „Space taxi"

svakako zaslužuje epitet izuzetno teške igre. U zavisnosti od vaših mogućnosti, odaberite vreme rada: jutro kao najlekše, dan, veče ili dežuranje svih 24 časa, kao najteže.



Komodor 64

SPARE CHANGE

Škrta razmena

Nosioci glavnih uloga u ovoj simpatičnoj igri su dva nestašna kradljivca i vaša skromna pojava. Možda je grubo za njih dvojicu reći da su kradljivci — nazovite ih kako hoćete, ali imajte na umu da oni „posuđuju" vaš novac i pedantno ga smeštaju u svoju kasicu-prasicu. U čemu je vaš problem? Vi biste rado otišli da gledate „show", ali je ulaznicu neophodno platiti u „sitnini", a vi imate samo zlato. Stoga krećete u akciju usitnjavanja zlata, i to prvo u gotovinu koja šuška (papirnate), a zatim i u sitne novčiće. Neophodno je prvo unovčiti zlato na kasi za papirnate novčanice, a zatim na posebnim punktovima za razmenu dobiti odgovarajuću sitninu, koju odlažete na posebno mesto. U međuvremenu, parokradice ispoljavaju izluđujuću ljubav prema vašem novcu. Ali, imaju i oni svojih slabih strana — dovoljno je da se odreknete jednog novčića i ubacite ga u džu-boks: oni će se oduševljeno okupiti oko njega i jedno vreme đuskati u ritmu muzike za ples, ostavljajući i vas i vaš novac na miru. Kada popunite odgovarajući kvadrat novcem, vrata vam se otvaraju, a uz neznatne napore (još sitnine) prelazite u sledeći nivo. Tu vam na raspolaganju stoje dva telefona, a kasnije i aparat za kokice, kojima možete zaokupljati pažnju gramzljivaca. Sve u svemu, veoma dopadljiva igra sa već uobičajeno dobrom grafikom i pratećim zvucima (svirka na džu-boksu, brbljanje na telefonu . . .)

Komodor 64

DALLAS QUEST

Istraga u Dalasu

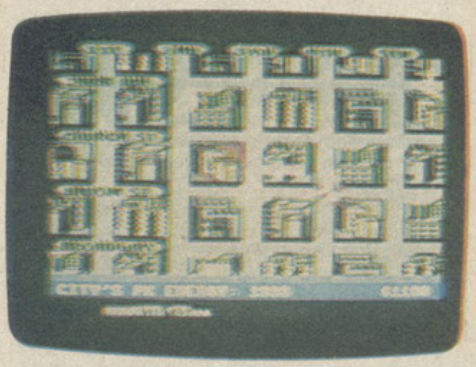
Na vaš bogat igrački meni stiže još jedna igra sa kratkim opisom: odlična avantura. Kao i većina avantura, i ova će vas, svakako, okupirati danima sve dok je potpuno ne osvojite. Kao detektiv, treba da rešite jedan komplikovan slučaj. I pored iskustva u poslu, ubrzo ćete (uglavnom prekasno) utvrditi da niste naučili da plivate ili jašete, kao, recimo, i to da nikada ne treba brzopleto izlaziti kroz glavna vrata. Ono što će, svakako, obradovati svakog igrača avantura je ogroman broj gotovo fotografski urađenih slika (oko 100), pa čak i zvuk (!!). Rečnik je izuzetno bogat i velik i nije baš nužno da komunicirate „tarzanskim jezikom", jer su moguće i dosta složene rečenice. Igra zauzima čitavu disketu (170 K!), pa se zato ne treba čuditi što je razrađena do najsitnijih detalja. Uglavnom, budite sigurni da ćete na svako ispitivanje dobiti opširan odgovor, a ne „I see nothing special!", kao u mnogim drugim avanturama. Sve u svemu, ova avantura je izuzetno zanimljiva, ali i teška za igranje, jer je potrebno imati zaista detektivski nos i dobru intuiciju da uzimate odgovarajuće (često banalne) predmete u pravo vreme (npr. rog čijim ćete zvukom kasnije skrenuti stampedo koji juri pravo na vas).

Komodor 64

GHOSTBUSTERS

Isterivači duhova

Ne verujete u duhove? Svejedno, pokušajte da se oprobate kao jedan ako ne baš vrhunski, ono bar prosečni istrebljivač dotičnih „malih belih". Očigledno im iz prve ne delujete baš previše autoritativno, jer ćete na početku same igre čuti dosta podsmešljiv uzvik „Ghostbusters! Ha! Ha! Haaa!" — izuzetno čisto, i razumljivo izgovoren. Virtuozno sviranje istoimene pesme oduševiće svakoga, a ona će vam biti jedini verni pratilac kada krenete u istrebljivanje. Cilj vam je, inače, da zaradite što više para . . . hm! možda je bolje reći cilj vam je da na, oduševljenje građana, spasete grad od duhova i „usput" zaradite koji dolar. Vaš lov počinje tako što vam banka pozajmljuje 10000 dolara za opremu koju treba da poseduje svaki „duholovac" — kola, zamke za duhove, detektor za duhove, usisivače za duhove i sl. Na mapi grada označićete svoju putanju i, zatim, krenuti u susret duhovima koji se drsko šetaju ulicama. Tom prilikom razlikujete obične duhove (slimers) i zle (marshmalow men). Istina, oni su mnogo ređi, ali vas mogu stajati dosta napora i para. Za razliku od slimersa, oni su sposobni čak i da uništavaju zgrade, a nastalu štetu plaćate vi ukoliko ih u dogledno vreme ne zarobite specijalnom zamkom. Na centru ekrana nalazi se palata Zuul, tj. centar duhova koji predstavlja vaš krajnji zadatak — mesto konačnog obračuna. Ukoliko na kraju igre imate više para nego na početku, dobijate račun sa kojim možete igrati sledeći put. U suprotnom, vaš neuspeh biće propraćen smehom.

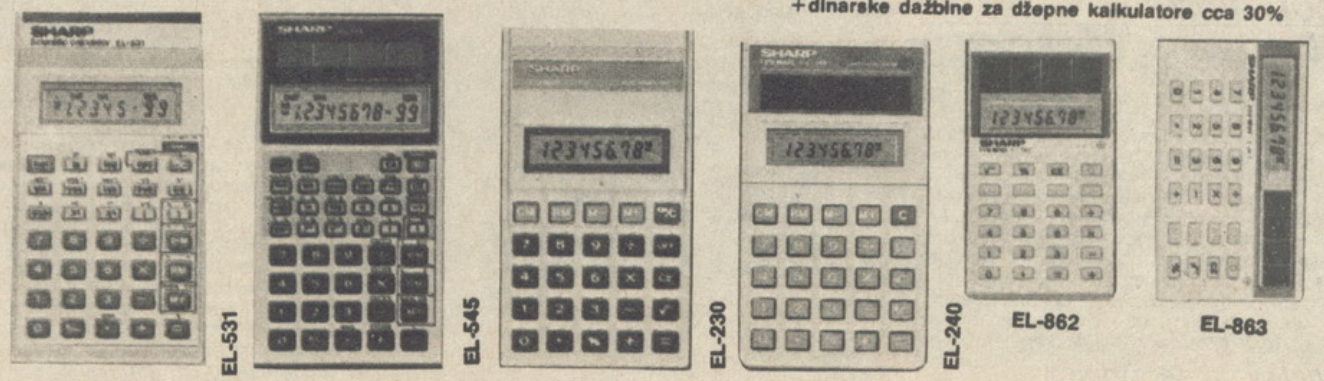


SHARP ZASTUPA I PRODAJE: TOZD CONTAL
Titova 66
61000 Ljubljana
Telex: 31175
Informacije na telefon: 061/328-441

„SHARP" džepni kalkulatori:

EL 230 (13 DM), solarni kalkulatori EL 240 (18 DM) i EL 862 (27 DM)
EL 863 (27 DM), profesionalni kalkulatori EL 531 (30 DM) i EL 545 (62 DM)

+dinarske dažbine za džepne kalkulatore cca 30%



zablude oko „elize"

Računari i ljudski um

Naglim uvozom računara počeli smo da kopiramo i određene pojave. Jedna od njih je i „sindrom još jednog koraka do inteligentnog računara". Svaki put kad neko na Istoku ili Zapadu izjavi da je izdvojio toliko i toliko dolara za dalji rad na razvoju „inteligentnog računara", javnost se uznemiri kao da nas od ostvarenja te želje deli još samo nedelju dana. Šta je dovelo do odomaćenja ovakvih uverenja među laicima? To je, nesumnjivo, brzi razvoj računara, ali i suštinsko nepoznavanje biti računarske nauke. Prve zablude oko računara nastale su kad i prvi računar, možda i pre, ali je ovaj tip zabluda najlakše objasniti zabludama koje se javljaju oko jednog od prvih i najpoznatijih „kvaziinteligentnih" programa. Razume se, to je program „Eliza".

Program „Eliza" je sredinom šezdesetih godina napravio Džozef Vajzenbaum, profesor na MITu (Massachusetts Institute of Technology). Taj antologijski program, iako bi se moglo pomisliti da je odavno zastareo, predstavlja model za većinu programa koji simuliraju dijalog sa korisnikom. Najsvežiji „potomak" „Elize" je program „Therapy" Stivena Rubia. Zablude vezane za „Elizu" još uvek su aktuelne u vezi sa svim „pričljivim" programima.

„Eliza" se sastoji iz opštih metoda za analizu rečenice i rečeničkih delova. Dejstvuje tako što prepoznaje „ključne" reči u tekstu, a zatim gradi na osnovu zadate rečenice potvrdnu rečenicu ili reaguje već zadatim pitanjem. Sastoji se iz dva nivoa — prvog, koji analizira jezik, i drugog, koji sadrži skup pravila koja joj govore kako konkretno treba da formuliše svoje rečenice. Ovaj drugi nivo je Vajzenbaum nazvao „scenario M" po kome „Eliza" igra. Pošto sama „Eliza" nema šta da kaže, „scenario" predstavlja skup pravila koji joj omogućava da improvizuje na osnovu podataka koje korisnik unosi preko tastature.

Živi procesor informacija

Prvi i najpoznatiji „scenario" koji je „Eliza" dobila da igra je bila uloga psihoterapeuta, i to ne bilo kog psihoterapeuta već psihoterapeuta rodžerovske orijentacije, pošto takva uloga zahteva samo da se sagovorniku kao reakcija vraćaju njegovi iskazi, malo transformisani. Takva „Eliza" je stekla ogromnu popularnost, pod imenom „Doktor". Naravno, sa širokim prihvatanjem „Elize" (i DOKTORA) na tržištu, došlo je i do pogrešnih tumačenja moći i kvaliteta tog programa. Sam Vajzenbaum je bio veoma iznenađen pogrešnim tumačenjem potencijala „Elize". To ga je i navelo da izvestan broj godina posveti proučavanju nauka koje mu nisu bile preterano bliske (sociologija itd . . .) i da napiše izvanrednu knjigu, „Moć računara i ljudski um", koja je i kod nas izašla u izdanju „Rada". U njoj on na najbolji mogući način objašnjava i raščlanjuje zablude nastale povodom „Elize".

Prvu zabludu je predstavljalo verovanje da „Doktor" („Eliza" sa svojim prvim sce-

— Vidiš? Moj nekadašnji seksepil još me nije napustio!

nariom) može postati osnova za razvoj potpuno automatizovanog oblika psihoterapije. Što je najčudnije, najvatreniji pobornici tog shvatanja su bili iz branše psihologa. Dr. Kenet Kolbi, Sajmon, Njuvel i drugi autori smatraju da postoji analogija između čoveka i računara i da se i čovek može shvatiti kao procesor informacija. Na osnovu toga tvrde da bi opšti metod za rešavanje problema (GPS) mogao da se primeni i na čoveka. To bi značilo da čoveka posmatramo kao sistem čije stanje se razlikuje od željenog koji se može „uraditi" unošenjem podataka i direktiva: sistem u željeno stanje.

Vajzenbaum tvrdi da je analogija između čoveka i računara ispravna samo na određenom nivou uprošćavanja čoveka. Glavna zamerka je u tome što zamišljajući čoveka kao procesor informacija apstrahujemo čitav niz karakteristika koje predstavljaju suštinske kvalitete njegove ličnosti, to jest upravo onaj deo njegove ličnosti koji mi pokušavamo da izlečimo. Sledeća zamerka je da kruti mehanizam „rešavanja problema" ne sadrži etiku neophodnu za „kontakt koji isceljuje" jer insistira na pragmatičnosti. Vajzenbaum ne raspravlja o tome da li je izvodljivo (tehnički) da računar dejstvuje kao psihoterapeut — on tvrdi da računar, u svakom slučaju, to **ne sme** da radi.

Muke sa jezikom

Sledeći nesporazum oko „Elize" je bilo rasprostranjeno mišljenje da ona demonstrira opšte rešenje za problem računarskog razumevanja prirodnog jezika. Sa uvek aktuelnom željom da se nešto učini na stvaranju „veštačke inteligencije", jasno je da je jedan od glavnih problema mašinsko razumevanje prirodnog jezika. Pošto svaki program u mašini definiše neki mikrosistem za čije rešavanje računar tako postaje stručnjak, ideja je da se, savladavši računarsko razumevanje prirodnog jezika, u kompjuter direktnom komunikacijom unese kompletno ljudsko znanje i tako računar oformi kao opšti Rešavač Problema. Naravno, prethodno treba ukloniti mnogobrojne probleme kao što su pitanje memorije, složenosti procesorskog dela i razumevanja prirodnog jezika. Ali, šta ako su neki od tih problema nerešivi, ili barem delimično nerešivi?

Osnovna ideja Šenka, Milera i drugih je bila da se definiše pojmovna struktura koja je interjezička i na koju bi se preslikavale jezičke strukture prirodnog jezika u toku procesa razumevanja. To bi omogućilo struktuiranje računara tako da on može da razume prirodni jezik. Ali, to nije tako jednostavno kao što zvuči. Vajzenbaum, s pravom, prigovara da je zanemarena višeslojnost prirodnih jezičkih struktura i da zbog toga svi primeri mogu da dejstvuju samo na veoma ograničenom mikrosistemu. To važi i za „Elizu". „Eliza" samo simulira razgovor — njeno razumevanje dijaloga jednako je razumevanju koje bi mogao da ima neki papagaj. Sama „Eliza" ne donosi ništa novo u željenom pravcu — da se formuliše program konačnog obima koji bi pridruživao pojmove strukture beskonačnom skupu rečenica koje se mogu formirati u prirodnom jeziku.

U svojoj knjizi „The Dragons of Eden" Karl Sagan ne samo što tvrdi da je „Eliza" konstruisana kao psihoterapeutski program nego i kaže da ne navodi kvalitete „Elize" u odnosu na ljudske psihoterapeute zato da bi nipodaštavao psihijatriju kao profesiju već da bi najvaio dolazak mašinske inteligencije. Očito je koliko i privid „inteligentnog" razgovora može da zavara čak i nekog ko ima naučno obrazovanje.

„Ne vidim ograničenja u inteligenciji koja se može dati računaru", rekao bi Džozef Vajzenbaum, „ali bez obzira na to, ta inteligencija će uvek biti sasvim drugačije prirode od ljudske."

Branko Đaković

BIBLIOTEKA

prosveta

Andrija Kolundžić UPOZNAJTE SVOJ RAČUNAR

Carinske rampe su dignute i vi ste nabavili svoj prvi računar. I taman ste pomislili da su svi vaši problemi rešeni — kad, ono, čarobna kutija se ne ponaša nimalo onako kako biste vi to želeli. Opire se, jogunasta je, ne razume vas. Ali samo zakratko. Vodeći znalac kompjuteristike, Andrija Kolundžić, pomoći će vam da ukrotite svoju kompjutersku goropad, da je potčinite svojoj volji, da najzad otvorite njene čarobne dveri. Bez obzira na to da li ste potpuni početnik ili već iskusni „haker" ovo je knjiga bez koje nipošto ne biste smeli da zalazite na opasnu teritoriju kompjutera.

Format 11,5×21 cm, strana 160, povez broširan, latinica

PRETPLATNA CEN 520.— DINARA
IZLAZI IZ ŠTAMPE POLOVINOM 1985. GODINE!

Dž.Dž. Ajsenk KNJIGA INTELIGENCIJE

Ova knjiga vam neće pomoći da postanete pametniji. Ali će vam zato biti od koristi u svakom drugom pogledu. Zabavićete se, pre svega, rešavajući domišljate testove koje je za vas priredio danas nesumnjivo vodeći svetski stručnjak za ljudsku inteligenciju; uz to, stavićete sebe na probu, prihvativši izazov ogledanja u pronicljivosti i veštini logičkog zaključivanja; konačno, ako spadate u one koji se ne ustežu ni od kakvih tajni, tu je posebni blok namenjen „intelektualnim divovima", sasvim dostojan i jednog Ajnštajna.

Format 11,5×21 cm, strana 160, povez broširan, latinica

PRETPLATNA CENA 520.— DINARA
IZLAZI IZ ŠTAMPE U DRUGOJ POLOVINI 1985. GODINE!

PRETPLATOM NA NAŠA IZDANJA OBEZBEĐUJETE SVOJ PRIMERAK POD NAJPOVOLJNIJIM USLOVIMA!

Potrebno zaokružiti
RAČUNAR/5, april 1985.
Porudžbeni kupon 906

_____
(Mesto i datum)

IRO „PROSVETA" OOUR Izdavačka delatnost
11001 Beograd, Dobračina 30
Poštanski fah 555, telefon 642-772

Pretplaćujem se na:
— A. Kolundžić UPOZNAJTE SVOJ RAČUNAR ..... 520.—
— Dž.Dž. Ajsenk KNJIGA INTELIGENCIJE ..... 520.—

Iznos od _____ dinara, obavezujem se isplatiti odmah, u roku od 8 dana, po prijemu „Prosvetine" uplatnice.

U slučaju spora ugovorena je nadležnost V opštinskog suda u Beogradu.

_____
(Prezime, ime oca, ime i zanimanje)

_____
(Naziv, adresa i telefon radne organizacije)

_____
(Broj pošte, mesto, adresa i telefon u stanu)

_____
(Svojeručni potpis, br.l.k./SUP)

Knjige slati na: STAN — RADNU ORGANIZACIJU

SAČEKAJTE „PROSVETINU" UPLATNICU!
NEPOTPUNE PORUDŽBENICE NEĆEMO UZIMATI U OBZIR!

KNJIGE ZA VAŠ USPEH U ŽIVOTU...

BIBLIOTEKA SAZNANJA

1. **A.M. Semorle: „I VI MOŽETE IMATI SUPER PAMĆENJE"**

— VI izd. Programirani priručnik tehnike pamćenja i učenje, kojim možete višestruko povećati svoje pamćenje i sposobnost učenja.

I knjiga: **„METODE I TEHNIKE PAMĆENJA"** (202 strane) Din 800.-
II knjiga: **„PRAKTIČNA PRIMENA"** (228 str.) — Din 800.-

PRAKTIČNE, LAKO SHVATLJIVE I SVIMA PRISTUPAČNE METODE **BRZOG, LAKOG I TRAJNOG UČENJA I PAMĆENJA svih vrsta podataka i znanja.**

Komplet (obe knjige zajedno) Din. 1.500.-
Format 17×24 cm, 430 str., preko 150 ilustracija, 15 samotestova.

2. **S. Yesudian — E. Haich: „JOGA I SPORT"**

— Kompletan, praktičan i veoma pristupačan priručnik HATHA JOGE, drevne i **širom sveta potvrđene** jedinstvene umetnosti istoka **za sticanje i očuvanje dobrog zdravlja tokom celog života.**

Ko želi **DOBRO ZDRAVLJE I LEPU LINIJU, PSIHIČKI MIR I SNAGU** treba da prouči ovu veoma **rasprostranjenu knjigu prevedenu na 19 jezika sveta** u preko dva miliona primeraka.

Cena Din. 800.- džepni format, 224 str., preko 70 ilustracija.

3. **S. Yesudian: „JOGA I ZDRAVLJE"**

— Viši kurs HATHA JOGE, nastavak knjige „„JOGA I SPORT". **Programirani priručnik primene najboljih metoda Hatha joge za svaku sedmicu u godini.**

SPECIJALNI REGISTAR I DIJAGNOSTIČKI INDEKS za preko 140 oboljenja omogućuju **efikasnu primenu najboljih metoda za pojedine organe i oboljenja. Cena** Din. 800.- džepni format, 221 str., preko 250 ilustracija.

Komplet (obe knjige „JOGA I SPORT" I „JOGA I ZDRAVLJE" zajedno) Din. 1.500.-

Sve knjige su štampane latinicom, na finom ofset papiru sa koricama u više boja.
GARANCIJA: Svakom nezadovoljnom čitaocu naših knjiga odmah vraćamo novac, ako neoštećenu knjigu vrati u roku oc pet dana nakon prijema.
Kod isporuka za inostranstvo zaračunavamo poštarinu i pakovanje.

BIBLIOTEKA SAZNANJA

NARUDŽBENICA — RAČUNAR V

1. **SUPER PAMĆENJE** _____ kom.
2. **JOGA I SPORT** _____ kom.
3. **JOGA I ZDRAVLJE** _____ kom.

Adresa naručioca (ispuniti čitko, ŠTAMPANIM SLOVIMA)

Ime i prezime _____

Ulica i broj _____

Pošt. br. i mesto _____

Knjige plaćate pouzećem — poštaru kod preuzimanja

Naručite na adresu: „BIBLIOTEKA SAZNANJA" p.p. — 20/70, 11031 BEOGRAD 8.

partizanska knjiga

LJUBLJANA
OOUR IZDAVAČKO-PUBLICISTIČKA DELATNOST
BEOGRAD

objavljuje pretplatu na delo

KOMPJUTERI SVET BUDUĆNOSTI

MALA ENCIKLOPEDIJA

Priredio:
prof. dr Miroslav Pečujlić

Urednik:
Aleksandar Postolović

prvi deo:

NOVA SLIKA SVETA
(DVA LIKA POSTINDUSTRIJSKE CIVILIZACIJE)

A. **TEORIJSKI ESEJI**

— **Anatomija postindustrijske revolucije**
(Kompjuterska revolucija i društvo informatike — industrija gena)
— **Alternativna tehnologija i humanije društvo**
— **Između tehnološke utopije i tehnofobije**
(„Tehnika rešava sve" — „Spasite nas od tehnike")
— **Tehnološka moć i sloboda — ili podvlašćivanje**
(Humanizacija rada ili novi oblici otuđenja)
— **„Industrija smrti"**
— **Ekologija i tehnologija**
— **„Svetski grad i svetsko selo"**
(Perspektiva Trećeg sveta — Novi razvoj ili tehnološki kolonijalizam)
— **Protagonisti budućnosti**
(Socijalne sile koje kontrolišu poslednju fazu industrijalizma ili oblikuju dolazeću civilizaciju)
— **Inteligentne mašine i čovek**
„Kompjuterske generacije"
(Robotizovani humanoidi ili nova individualnost)
— **Kompjuteri i socijalizam**
(Postindustrijski tehnokratski kapitalizam)
— **Autoritarni socijalizam**
(Mogućnosti slobodarskog socijalizma)
— **Jugoslovensko društvo**
(Puko kopiranje ili i vlastito stvaralaštvo; Informatika — racionalno upravljanje i samoupravljanje)
— **Istorijat jugoslovenske atomistike**

Pisci teorijskih priloga su autori: akademik Aleksandar Despić, prof Vlastimir Matejić, prof. Vladimir Milić, prof. Miroslav Pečujlić, akademik Pavle Savić, dr Miloš Sinđić, prof. Radmila Stojanović, prof. Vladimir Štambuk, prof. Rajko Tomović, prof. Zoran Vidaković.

B. **„SVETSKI OKRUGLI STO" KALEIDOSKOP PROMENA KRAJEM XX VEKA**

(Fragmenti iz dela najznačajnijih svetskih pisaca)

— A. Tofler: „Kaleidoskop promena"
— E. E. Šrajber: „Svetski izazov"
— D. Naisbett: „Megatrendovi"
— D. Bel: „Dolazak postindustrijskog društva"
— A. Turen: „Programirano društvo"
— D. K. Galbrajt: „Doba neizvesnosti"
— A. Gorc: „Tehnika, tehničari i klasna borba"
— Dž. Orvel: „1984"
— J. Habermas: „Komunikativno delanje"
— A. Gouldner: „Od ideologije ka tehnologiji"
— Y. Masuda: „Kompjuto-utopija"

Pripremio
dr Vladimir Milić

drugi deo:

TEHNOLOGIJA I SOCIOLOGIJA

A. **POLJA PRIMENE**

— Automatizovana fabrika
— Industrijski roboti
— Elektronska kancelarija
— Računari u kući (lični kompjuter)
— Telekomunikacije — Elektronska pošta
— Video-revolucija
— Kompjuteri u obrazovanju
— Umetnost i kompjuteri
— Medicinski kompjuter
— Elektronika u bankama
— Kompjuteri u trgovini
— Kompjuteri u saobraćaju
— Elektronika i urbanizam
— Informatika i moderna poljoprivreda
— Kompjuterizovan kriminal

Pripremio
dr Vladimir Štambuk

B. **IZAZOVI — EMANCIPACIJA ILI DEHUMANIZACIJA**

— Razvoj ljudskih sposobnosti ili masovna nezaposlenost
— Kriza i budućnost rada (kreativni rad ili dekvalifikacija radne snage)
— Kompjuteri i nova ekonomija
— Kompjuteri i demokratija ili totalna kontrola

Pripremio
dr Predrag Radenović

★★★

treći deo:

ŠTA JE KOMPJUTER? ANATOMIJA KOMPJUTERA

— Evolucija kompjutera — Od prvih računskih mašina do superkompjutera
— Razvoj Hardwer-a
— Razvoj Softwer-a
— Kompjuteri pete generacije — Veštačka inteligencija
— Programiranje
— Kompjuterski jezik

Pripremio
Zoran Mastilović

★★★

četvrti deo:

KOMPJUTERSKI REČNIK
(MALI LEKSIKON)

Sadržaj drugog, trećeg i četvrtog dela sastoji se prevashodno iz prevoda i prikaza radova istaknutih svetskih autora.
Knjiga sadrži 740 strana velikog formata; opremljena je velikim brojem fotografija u koloru i grafičkim prilozima. Povez tvrd sa omotom. Knjiga će izaći krajem 1985.
Prva cena u pretplati: 4.500 din. Cena u prodaji biće znatno veća.

NARUDŽBENICA — R_5

Partizanska knjiga — Beograd,
Bulevar vojvode Mišića 17/VII — tel. 651-672, 650-297

Ovim neopozivo poručujem delo KOMPJUTERI — SVET BUDUĆNOSTI (mala enciklopedija)

po prvoj pretplatnoj ceni od 4.500 din.

Ovaj iznos obavezujem se uplatiti:
— odjednom sa 20% popusta u iznosu od 3.600 dinara, ili

— u _____ mesečnih rata (najviše šest).

_____
Ime i prezime i ime oca

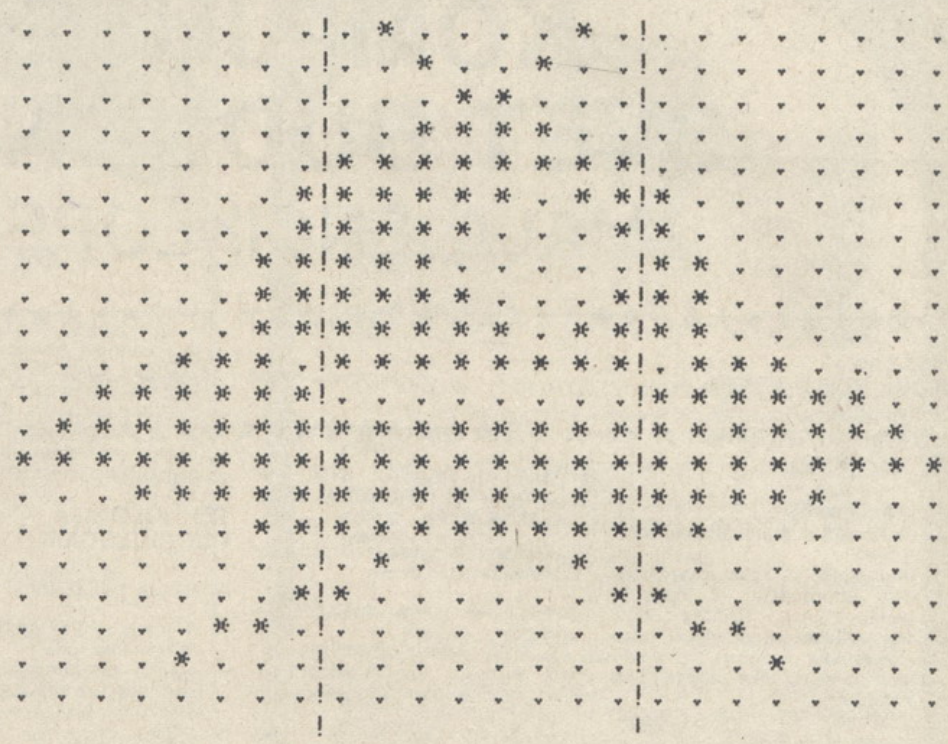
_____
Adresa stana, mesto, br. l. k. i od koga je izdata

_____
Naziv radne organizacije kod koje je kupac zaposlen

_____
Overa radne organizacije

sprajtovi na „komodoru"

Programiranje u bejziku



„Komodor 64" sadrži u sebi potencijale koji omogućuju da pored glavne slike, sačinjene od 320 puta 200 tačaka, na ekranu, kojom prikazujete tekst ili grafiku, napravite još 8 malih slika od 24 puta 21 tačke koje će od ekrana biti potpuno nezavisne. Neće se prikazivati dok vi to ne zatražite, a kada to učinite svaka od njih iskrsnuće tačno na onom mestu na ekranu koje ste prethodno odredili. Ostaće tamo sve dok je ne isključite ili pomerite na neko drugo mjesto. Ovo još nije sve: možete jednostavno da kontrolišete da li su se dve slike dotakle, ili da li je neka dotakla osvetljenu tačku na glavnom ekranu. Pošto su nezavisne od glavnog ekrana, mogu se prikazivati tako da zaklanjaju njegov sadržaj, ili da njime budu zaklonjeni. Takođe, pošto su međusobno nezavisne, mogu zaklanjati i jedna drugu. Boje kojima su predstavljene takođe su nezavisne — u svakoj od njih može biti najviše četiri boje.

Uobičajen naziv za ovakve male slike je SPRAJT (SPRITE) i tako ćemo ih ubuduće i zvati. Pošto ih istovremeno može biti 8, označimo ih brojevima od 0 do 7.

Šta uraditi da se sprajt pojavi i pokrene?

Da dobijete sprajt na željenom mestu na ekranu, morate obaviti sledeće radnje:

1) Definisati izgled sprajta;
2) Postaviti njegov opis na neko mesto u memoriji;
3) Odrediti njegovu boju;
4) Postaviti koordinate mesta od koga počinje prikaz;
5) Uključiti prikazivanje.

Definisanje izgleda

Ova etapa je jedina u kojoj računar ne može mnogo pomoći. Prepušteni ste svojoj mašti i smislu za grafički prikaz. Tehnika je jednostavna: nacrtajte polje od 21 reda po 24 tačke i u njemu pojačajte svaku tačku za koju želite da bude deo slike. U prikazu sprajta samo će se ove tačke videti, dok će sve ostale biti prikazane u boji pozadine (osnovnoj boji ekrana). Kada završite, podelite polje na tri uspravne kolone po 8 tačaka.

Sada zamislite da su sve pojačane tačke prikazane kao jedan, a ostale kao nula. Izračunajte tako za svaku od 21 linija po tri dekadna broja koji predstavljaju zapis svake od 8 grupa nula jedinica u toj liniji. Na primer, za drugu grupu tačaka u prvoj liniji biće

$$01000010 = 2^6 + 2 = 66$$

Polje tačaka može se predstaviti brojevima na sledeći način:

0	66	0
0	36	0
0	24	0
0	60	0
0	255	0
1	251	128
1	241	128
3	224	192
3	241	192
3	251	192
14	255	112
63	0	252
127	255	254
255	255	255
31	255	248
3	255	192
0	66	0
1	129	128
6	0	96
8	0	16
8	0	0

Sa ova 63 broja sprajt je potpuno definisan i možete preći na sledeću fazu.

Postavljanje opisa u memoriju

Da bismo ovaj ovaj postupak učinili što jasnijim, definišimo na početku sledeće pojmove:

1) Adresa sprajta — ova adresa nije prava memorijska adresa, već adresa grupe od 64 registra kojima se zapisuje definicija sprajta u memoriji. Kako je za opis potrebno 63 mesta, 64-to se ne koristi. Adresa sprajta može biti između 0 i 255.

2) Adresa prvog registra u opisu sprajta — ova adresa je prava memorijska adresa i ukazuje na prvi broj u definiciji sprajta. Dobija se tako što se adresa sprajta pomnoži sa četiri.

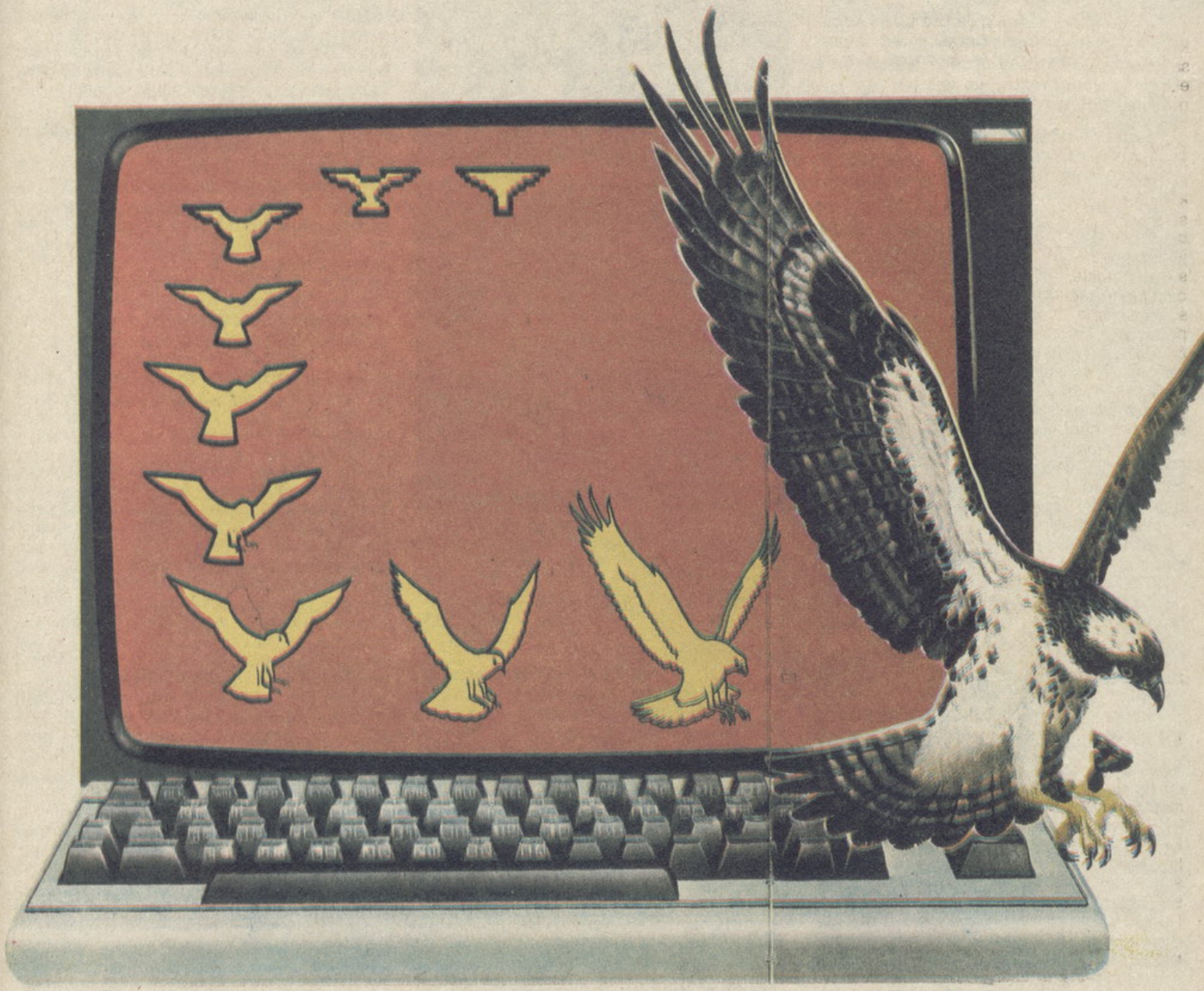
Vašem računaru će biti jasno kako sprajt izgleda tek onda kada bude znao gde se u memoriji nalazi njegov opis. To morate saopštiti za svaki sprajt koji želite da koristite postavljanjem adrese sprajta u za to predviđeni registar:

za sprajt 0	registar 2040
za sprajt 1	registar 2041
za sprajt 7	registar 2047

Nakon toga, počev od adrese prvog broja u opisu sprajta, upišite jedan za drugim 63 broja, uzimajući ih vrstu po vrstu. Time ste završili postupak postavljanja opisa sprajta u memoriju.

Pri tome se ne mogu koristiti adrese sprajta od 0 do 31 (izuzev 13, 14, 15), jer se na njima nalaze važne sistemske promenljive — one bi bile uništene a program prekinut. Takođe, ako koristite adresu sprajta 32 ili nekoliko narednih, doći će do preklapanja programa i opisa sprajta, pri čemu će program biti uništen. Zato je dobro kao ove adrese koristiti brojeve veće od 128 u modu za tekst ili nešto niže od 128 u modu za visoku rezoluciju.

Naučili ste kako da postavite tačku na ekran vašeg „komodora 64". Nacrtali ste i prvu sliku — na primer, mali leteći tanjir. Želite sada da napravite takav program koji će se pokretati po ekranu i spustiti na neku zaravnjenu površinu. Kako pomeriti sliku na ekranu? Treba celu grupu tačaka premestiti iz jednog dela ekranske memorije u drugi, ali tako da „niko ništa ne primeti". U bejziku se, odmah da kažemo, ovaj uslov ne može ispuniti. Ako ste vični mašinskom programiranju, uložićete svo svoje znanje da ostvarite ovaj efekat, ali nismo sigurni da ćete biti zadovoljni rezultatom. Glavu gore! Jedan način ipak postoji.



Postavljanje boje sprajta

Boje su kod C64 kodirane brojevima od 0 do 15. Za svaki sprajt postoji njegov registar boje i boja sprajta određuje se upisom koda boje u taj registar.

za sprajt 0	registar 53287
za sprajt 1	registar 53288
za sprajt 7	registar 53294

Postavljanje koordinata sprajta

Pod koordinatom sprajta podrazumeva se koordinata gornje leve tačke sprajta na celom ekranu. Koordinate sprajta nisu isto što i koordinate vidljivih tačaka (320×200). Naime, koordinate tačaka obuhvataju samo vidljivi deo ekrana, dok koordinate sprajtova mogu biti i van tog dela, u ramu slike. Tada se sprajt ne vidi. Na taj način omogućeno je da sprajt može da ulazi i izlazi iz vidljivog dela, pri čemu se, naravno, vidi samo onaj deo koji je u vidljivom polju.

Evo kakva veza postoji između koordinata sprajtova i koordinata vidljivog dela ekrana:

koordinate tačke u vidljivom delu		koordinate tačke na celom ekranu	
X	Y	X	Y
0	0	24	50
319	0	343	50
0	199	24	249
319	199	343	249

Odavde se može zaključiti (imajući u vidu da sprajt ima 24×21 tačku) da je sprajt delimično vidljiv ako su obe njegove koordinate u sledećim granicama:

1 =< X =< 343
30 =< Y =< 249

Dakle, transformacija vidljivih koordinata u ekranske vrši se po sledećim formulama:

X=VX+24 Y=VY+50

U jedan registar od 8 bita može biti upisan jedino broj između 0 i 255. Za zapis Y koordinate to je dovoljno, ali za X nije. Zato je za zapis dodatnog bita predviđen registar 53264. Bit O ovog registra služi za upis dodatnog bita X-koordinate sprajta 0, bit 1 za dodatni bit X-koordinate sprajta 1 i slično za ostale.

Ovakav zapis zahteva da pre upisa koordinate obavite sledeće operacije:

1) celobrojno deljenje vrednosti X-a sa 256

2) upis ostatka u registar za niži deo vrednosti X-a

3) upis rezultata deljenja u odgovarajući bit registra za dodatne bitove.

Dakle, postupak je sledeći:

```
DBIT=INT (X/256)
POKE XSPR,X-256*DBIT
IF DBIT=0 THEN POKE 53248, PEEK (53264) AND (255-2 BRSPR)
IF DBIT=1 THEN POKE 53248, PEEK (53264) OR (2 BRSPR)
POKE YSPR,Y
```

(DBIT=dodatni bit za X)
(BRSPR=broj sprajta)
(XSPR=X-koo sprajta)
(YSPR=Y-koo sprajta)

Registri za koordinate su u memoriji raspoređeni na sledeći način:

X-koo sprajta 0	53248
Y-koo sprajta 0	53249
X-koo sprajta 1	53250
Y-koo sprajta 1	53251
X-koo sprajta 7	53262
Y-koo sprajta 7	53263
registar za dopunski bit	53264

Ovim se potpuno određuje pozicija sprajta na ekranu. Međutim, on se neće videti sve dok ne izvršite . . .

Uključivanje sprajta

Registar na adresi 53269 namenjen je uključivanju i isključivanju svih osam sprajtova. Bit 0 odgovara sprajtu broj 0, bit 1 sprajtu 1 . . . Postavljanjem nekog bita na 1 vrši se uključivanje sprajta, t.j. njegov prikaz, a postavljanjem na 0 vrši se isključivanje, odnosno gašenje sprajta.

registar za uključivanje sprajtova 53269

Kretanje sprajta

Kada se sprajt jednom definiše, njegovo je pokretanje vrlo jednostavno: potrebno je jedino da u registre za X-koordinatu (niži i dodatni) i Y-koordinatu postavite nove vrednosti i sprajt će gotovo trenutno biti premešten sa starog na novo mesto. Pri tome se obavezno treba koristiti operacijama navedenim u postupku upisa početnih koordinata sprajta.

Odnos sprajta i pozadine

Sprajtovi mogu biti korišćeni u isto vreme kad i glavni ekran, i to na više načina:

Lična karta sprajtova

Adrese sprajtova 0 do 7	2040 do 2047
X-koo sprajta 0	53248
Y-koo sprajta 0	53249
X-koo sprajta 1	53250
Y-koo sprajta 1	53251
X-koo sprajta 2	53252
Y-koo sprajta 2	53253
X-koo sprajta 3	53254
Y-koo sprajta 3	53255
X-koo sprajta 4	53256
Y-koo sprajta 4	53257
X-koo sprajta 5	53258
Y-koo sprajta 5	53259
X-koo sprajta 6	53260
Y-koo sprajta 6	53261
X-koo sprajta 7	53262
Y-koo sprajta 7	53263
Dopunski bitovi za X-koo	53264
Uključivanje	53269
Vertikalno proširenje	53271
Prioritet sprajt-pozadina	53275
Višebojni sprajtovi	53276
Horizontalno proširenje	53277
Sudar sprajt—sprajt	53278
Sudar sprajt—pozadina	53279
Boja 01 višebojnog sprajta	53285
Boja 11 višebojnog sprajta	53286
Boja sprajta 0	53287
Boja sprajta 1	53288
Boja sprajta 2	53289
Boja sprajta 3	53290
Boja sprajta 4	53291
Boja sprajta 5	53292
Boja sprajta 6	53293
Boja sprajta 7	53294

— da bi sprajt zaklanjao tačke na ekranu, treba postaviti njegov karakterističan bit u registru 53275 na 0 i tada sprajt ima viši prioritet u odnosu na pozadinu.

— da sprajt prolazi ispod vidljivih tačaka ekrana treba isti bit postaviti na 1 (prioritet pozadine)

— da li sprajt i ekran imaju bar jednu zajedničku tačku, možete saznati čitajući odgovarajući bit registra 53279. Njegov sadržaj se u slučaju dodira tehnički postavlja na 1 i tako ostaje sve dok ne pročitate taj registar. To znači da će jednom otkriven sudar sprajta i pozadine biti zapamćen i ako se oni potom udalje. Tek čitanjem registra se svi bitovi automatski postavljaju na 0 i omogućuju detektovanje sledećeg sudara.

Postupak se sprovodi sledećom naredbom:

IF PEEK 53279)<>0 THEN . . . . obrada tog slučaja

Uzajamni odnos sprajtova

Sprajt sa nižim brojem uvek ima viši prioritet od onog sa višim brojem. To znači da će se u slučaju preklapanja dva sprajta prikazati onaj sa nižim brojem.

Da li su se dva sprajta dotakla možete ustanoviti slično kao sa pozadinom. Registar 53278 automatski postavlja one bitove čiji brojevi odgovaraju brojevima sprajtova koji su učestvovali u sudaru. I ovaj se registar automatski briše nakon čitanja. Ispitivanje da li se neki sudar uopšte dogodio može se izvršiti naredbom

IF PEEK (53278)<>0 THEN . . . obrada tog slučaja

Proširivanje sprajtova

Ako vam je slika 24×21 suviše mala, za vas je predviđena mogućnost proširivanja sprajtova. Ovo je moguće obaviti na tri načina:

1) povećanjem širine dva puta (na 48×21 tačku)

2) povećanjem visine dva puta (na 24×42 tačke)

3) povećanjem i širine i visine po dva puta (48×42 tačke)

U svim ovim slučajevima važi sledeće: definisanje sprajta je nepromenjeno, t.j. njegov opis je isti kao i ranije, ali se svaka njegova tačka prikazuje sa po dve tačke po širini ako se vrši proširivanje, po visini ako mu povećavate visinu i kvadratom od 2×2 tačke ako se proširivanje vrši u obe dimenzije.

Kao indikatori da li se sprajt prikazuje obično ili prošireno služe registri

za horizontalno proširenje 000 53271
za vertikalno proširenje 000 53277

Kao i obično, ako u nekom bitu stoji 1, odgovarajući sprajt biće prikazan prošireno. Dakle, ako su kombinacije vrednosti u ova dva registra

bit u 53271	bit u 53277	efekat
0	0	običan prikaz
1	0	horizontalno proširenje
0	1	vertikalno prišireni je
1	1	proširenje u oba pravca

Višebojni sprajtovi

Do sada smo govorili o sprajtovima čije su sve osvetljene tačke prikazane u istoj boji. Žrtvujući polovinu prikazanih tačaka po horizontali, t.j. svodeći dimenzije sprajta na 12×21 tačku, moguće je u okviru jednog sprajta dobiti do četiri boje. Opis sprajta ne razmatra se tada tačku po tačku, već par po par tačaka. Svaki par određuje kojom bojom će tačka na tom mestu biti obojena. Jedna od tih boja je boja ekrana, druga je boja određena za taj sprajt i smeštena je u njegov registar za boju, a preostale dve boje se unapred predviđaju za svih 8 sprajtova i smeštaju se u posebne za to predviđene registre na adresama 53285 i 53286.

Određivanje boje tačke višebojnog sprajta postiže se na sledeći način:

kombinacija bitova u opisu		adresa boje
0	0	boja pozadine 53281
0	1	53285
1	0	registar za boju sprajta
1	1	53286

Sprajt će biti prikazan jednobojno, bojom iz njegovog registra za boju, sve dok se ne označi kao višebojni sprajt, što se postiže postavljanjem odgovarajućeg bita u registru 53276.

kontrola višebojnih sprajtova 53286

1. Postavljanje bita u registar vrši se naredbom

POKE adresa, PEEK (adresa) OR (2 bit)

2. Brisanje bita iz registra vrši se naredbom

POKE adresa, PEEK (adresa) AND (255-2 bit)

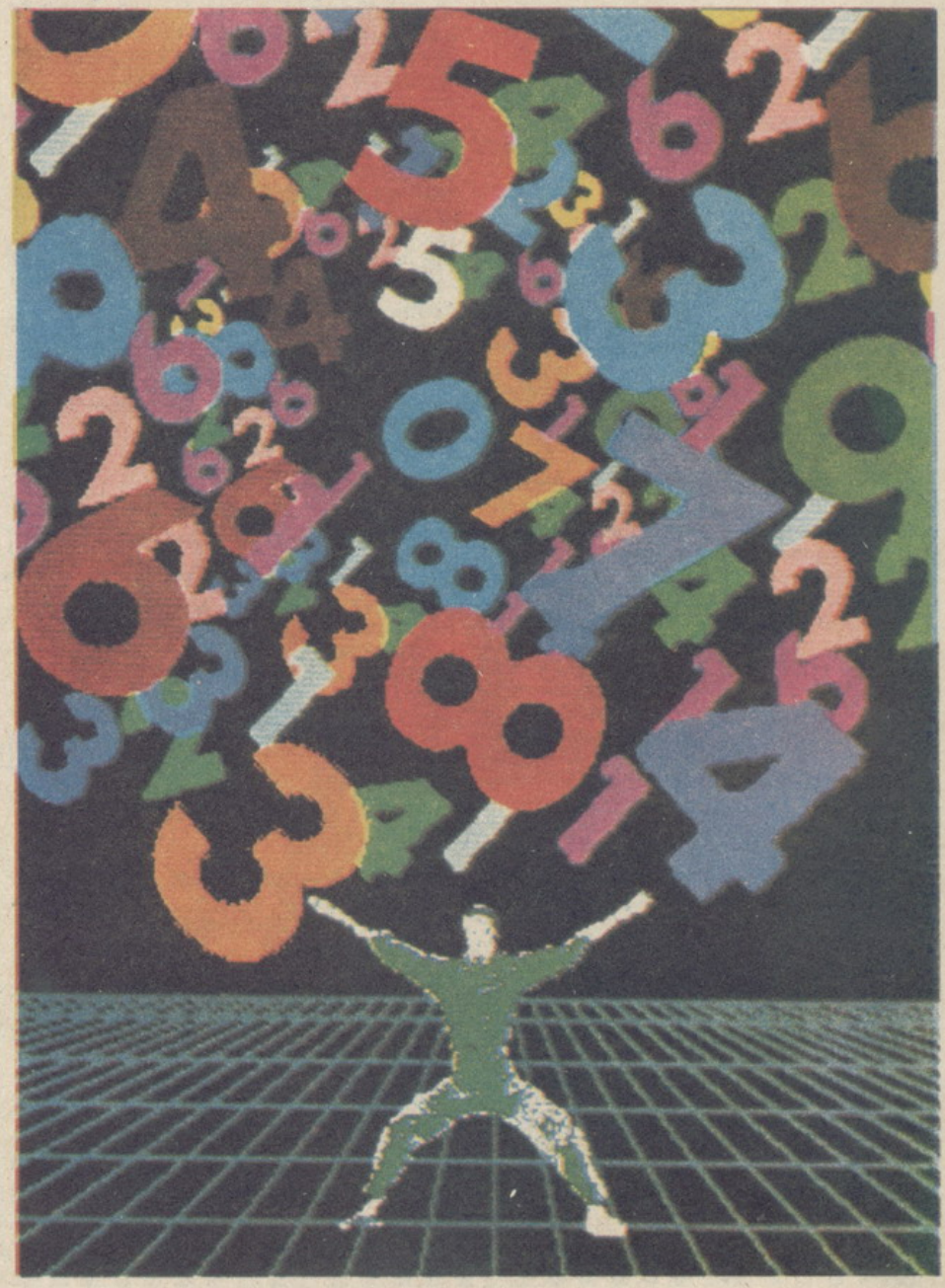
Duško Veljković

U svetu slučajnih brojeva

zec iz šešira

Programiranje u bejziku

Mnogi provode vreme otvarajući pasijans ili bacajući kockice, a poneki očekuju od sudbine da im jednim okretanjem ruleta pruži sve ono što drugi stiču mukotrpnim radom. Nepredvidivi obrti mogući su i u igrama na računaru zahvaljujući generatoru slučajnih brojeva. Ali „slučajno izabrani" brojevi mogu pružiti i više od igre. Puno knjiga posvećeno je oštroumnoj tehnici koja slučajne brojeve koristi za rešavanje složenih zadataka numeričke analize. Generisanje slučajnih vrednosti omogućava da se na računarima verno modeliraju prirodne pojave, a predstavlja i dobar izvor za dobijanje podataka pri testiranju različitih algoritama. Stoga je sposobnost računara da „izmišlja" slučajne brojeve znatno značajnija funkcija nego što se često misli.



Rasadnici brojeva

Generisanje slučajnih brojeva može da se realizuje na različite načine, ali najpraktičnije je da se koristi ugrađena sposobnost računara za vršenje aritmetičkih operacija Prvi takav pristup predložio je 1946. Džon fon Nojman koji je koristio „metodu sredina kvadrata". Njegova ideja se zasniva na tome da se prethodni slučajni broj kvadrira i da se, zatim, iz broja uzmu srednje cifre. Ako, na primer, radimo sa desetocifrenim brojevima i ako je prethodni broj bio 5772156649, njegov kvadrat iznosi 33317792380594909201, pa je stoga sledeći „slučajni" broj 7923805949. Metod odmah nameće pitanje da li postupak koji nije slučajan uopšte može da proizvodi slučajne brojeve. Odgovor je ne, ali nama i nije potreban slučajan niz brojeva, već niz koji izgleda kao da je slučajan, odnosno u kome ne možemo lako ustanoviti zakonitost po kojoj bi se svaki element mogao izraziti preko prethodnog. U naučno-tehničkoj literaturi nizovi brojeva do kojih dolazimo na determinističke načine nazivaju se pseudoslučajni ili kvazislučajni. Računari proizvode upravo pseudoslučajne brojeve koji samo gledano spolja nisu međusobno zavisni, ali su ipak dovoljno „slučajni" za mnoštvo aplikacija.

Svi kućni računari poseduju generatore slučajnih brojeva koji se iz bejzika pozivaju pomoću funkcije RND (RaNDom=slučajan). Ova funkcija proizvodi jedan slučajan broj iz intervala (Ø,1) ne uključujući Ø i 1. Da bismo videli te brojeve, unesimo sledeći kratak program. Razume se, kao i pre unošenja bilo kog novog programa, treba očistiti memoriju komandom NEW.

„Spektrum"

```
10LET X=RND
20PRINT X
30GOTO 10
```

„komodor" i „acorn"

```
10LETX=RND(1)
20PRINT X
30GOTO10
```

Ako imate „galaksiju", otkucajte program kao za spektrum, jedino u liniji 10 izostavite službenu reč LET. Po startovanju ovog programa, na ekranu će početi da promiče niz decimalnih brojeva koji ćete jedino „nasilno" moći da zaustavite: „spektrumu" treba pritisnuti STOP, pa ENTER, na komodoru RUN/STOP, na Acornu ESCAPE, a na „galaksiji" BRK.

U praksi nam obično trebaju celi brojevi iz nekog intervala, recimo brojevi od 1 do 6 ako želimo da simuliramo bacanje kockice. U tom slučaju, pozivamo u pomoć funkciju INT koja daje prvi ceo broj manji ili jednak vrednosti argumenta. Pazite, INT (INTeger — ceo broj) ne vrši zaokruživanje broja. Vrednost INT(1.9) nije 2, već 1 kao i da smo tražili INT(1.1). Tako brojeve od 1 do 6 možemo dobiti na sledeći način:

„spektrum"
```
10 LET X=INT(RND*6=1)
```

Acorn
```
10 LET X=RND(6)
```

„komodor"
```
10 LET X=INT(RND(1)*6+1)
```

„galaksija"
```
10 X=INT(RND*6+1)
```
Na sličan način možemo dobiti brojeve iz bilo kog intervala (vidi tablicu).

Tablica množenja

Iskoristimo ove činjenice da napravimo program za učenje tablice množenja. U „računarima 2" dali smo nekoliko rešenja ovog zadatka za računar „galaksija". Zadržimo se, sada, na pojednostavljenoj verziji ovog programa koja ima za cilj jedino da objasni korišćenje slučajnih brojeva i da nas uvede u problematiku programskih ciklusa, kojom ćemo se baviti u sledećem broju.

I ovom prilikom držaćemo se osnovnog pravila programiranja — pre pisanja programa treba da definišemo šta želimo njime da postignemo i kako ćemo to učiniti. Dakle, moramo prvo načiniti precizan plan rešavanja zadatka, pa tek onda preći na najlakši deo posla — samo pisanje programa.

Ideja je ovakva: da bi dete za koje sastavljamo program imalo utisak da računar s njim razgovara, najpre se od njega traži da unese svoje ime koje će se čuvati u promenljivoj X$ i kojim će mu se računar nadalje obraćati. Generisanje slučajnih brojeva omogućava nam da koliko god želimo puta postavimo problem „koliko je x puta y" uvek sa raznim brojevima x i y i na taj način pružimo detetu šansu da dobro uvežba tablicu množenja. Uzećemo da su x i y brojevi od 1 do 9, ali nema prepreke da se kasnije uzimaju i veći brojevi. Odgovor na pitanje koliko je x puta y računar će porediti sa rezultatom množenja i, ako je tačan, postavljaće detetu novo pitanje, a ako nije upućivati ga da ponovo reši isti zadatak.

„spektrum"
```
10 PRINT „ZDRAVO, KAKO SE ZOVEŠ?"
20 INPUT X$
30 CLS
40 PRINT „ZDRAVO, □"; X$; "IMAM NEKOLIKO PITANJA ZA TEBE"
50 PAUSE 200
60 CLS
70 LET N=INT (RND*9+1)
80 LETM=INT(RND*9+1)
90 PRINT „KOLIKO JE"; N; „PUTA"; M; „?"
100 INPUT A
110 IF A=N*M GOTO 150
120 PRINT N; „PUTA"; M; „=“; A; „???"
130 PRINT „POKUŠAJ PONOVO"
140 GOTO 90
150 PRINT „TAČNO, □"; X$; „EVO NOVOG PITANJA"
160 PAUSE 150
170 GOTO 60
```

„komodor"
```
10 PRINT „ZDRAVO, KAKO SE ZOVEŠ?"
20 INPUT X$
30 PRINT „SHIFT/CLR"
40 PRINT „ZDRAVO, □"; X$; „IMAM NEKOLIKO PITANJA ZA TEBE"
50 FOR X=1 TO 2000:NEXT X
60 PRINT „ SHIFT/CLR "
70 N=INT(RND(1)*9+1)
80 M=INT(RND(1)*9+1)
90 PRINT „KOLIKO JE"; N; „PUTA"; M; „?"
100 INPUT A
110 IF A=N*M GOTO150
120 PRINT N; „PUTA"; M; „=“; A; „???"
130 PRINT „POKUŠAJ PONOVO"
140 GOTO 90
150 PRINT „TAČNO, □"; X$; „EVO NOVOG PITANJA"
160 FOR X=1 TO 1500 NEXT X
170 GOTO 60
```

Acorn
```
10 PRINT „ZDRAVO, KAKO SE ZOVEŠ?"
20 INPUT X$
30 CLS
40 PRINT „ZDRAVO, □"; X$; „IMAM NEKOLIKO PITANJA ZA TEBE"
50 FOR X=1 TO 6000 : NEXT X
60 CLS
70 N=INT(9)
80 M=INT(9)
90 PRINT „KOLIKO JE"; N; „PUTA"; M; „?"
100 INPUT A
110 IF A=N*M THEN GOTO 150
120 PRINT N; „PUTA"; M; „=“; A; „???"
130 PRINT „POKUŠAJ PONOVO"
140 GOTO90
150 PRINT „TAČNO, □"; X$; „EVO NOVOG PITANJA"
160 FOR X=1 TO 4000:NEXT X
170 GOTO 60
```

„galaksija"
```
10 PRINT „KAKO SE ZOVEŠ?"
20 INPUT X$
30 HOME
40 PRINT „ZDRAVO, □"; X$; „IMAM NEKOLIKO PITANJA ZA TEBE"
50 FOR X=1 TO2000:NEXT X
60 HOME
70 N=INT(RND*9+1)
80 M=INT(RND*9+1)
90 PRINT „KOLIKO JE"; M; „PUTA"; N; „?"
100 INPUT A
110 IF A=M*N GOTO 150
120 PRINT N; „PUTA"; M; „=“; A; „???"
130 PRINT „POKUŠAJ PONOVO"
140 GOTO 90
150 PRINT „TAČNO, □"; X$; „EVO NOVOG PITANJA"
160 FOR X=1 TO 1500 NEXT X
170 GOTO 60
```

Kako zadati interval

RND na „spektrumu" i „galaksiji", a RND (1) na „komodoru"; Acorn generiše slučajne brojeve između 0 i 0.999999. Ako želite veći interval, treba da pomnožite odgovarajućim brojem originalnu RND funkciju. Na primer, ako su vam potrebni brojevi između 0 i 39.999999, pomnožićete originalnu funkciju sa 40.

Da biste odbacili decimale i dobili cele brojeve, upotrebite funkciju INT. Ako na prethodni primer upotrebite INT, dobićete cele brojeve od 0 do 39. Ako su vam pak potrebni brojevi od 1 do 40, treba izrazu dodati 1.

Formula:

INT (RND+p) za „spektrum" i „galaksiju"*

*INT (RND (1)*b+p) za „komodor" i Acorn daće vam neki od celih brojeva počev od broja p.*

Formula

*INT (B−A+1)*RND+A) za „spektrum" i „galaksiju"*

*INT (B−A+1)*RND (1)+A) za „komodor" i Acorn daće vam neki od celih brojeva iz intervala [A, B]. Ako imate Acornov računar, možete dobiti ceo broj iz intervala [1, X] navođenjem broja X kao argumenta RND FJE. Ova mogućnost ne postoji na druga tri računara.*

Program sa sto petlji

I ovaj program, poput prethodnih, nikada ne završava sa radom na prirodan način. To je primer tzv. beskonačne petlje ili ciklusa. Pod ciklusom podrazumevamo skup instrukcija programa koje se bar za neke vrednosti ulaznih promenljivih ponavljaju više od jedanput. U ovom programu imamo dva ciklusa — jedan čine instrukcije 60—170 i iz njega se ne može izaći na prirodan način, dok drugi, koji čine linije 90 do 140, predstavlja ciklus iz koga se može izaći sa linije 110 ako je dat tačan odgovor. Kriterijum po kome se izlazi iz ciklusa zove se izlazni kriterijum i prema njemu možemo cikluse podeliti na brojačke, za čiji opis u bejziku možete upotrebiti instrukcije FOR . . . NEXT, i one kod kojih kriterijum izlaska nije broj ponavljanja naredbi ciklusa, već zadovoljavanje nekog posebnog unapred definisanog zahteva, kao što je to, u našem slučaju, davanje tačnog odgovora. Da bismo mogli na regularan način da okončamo izvršavanje našeg programa, moramo u spoljašnjem ciklusu definisati izlazni kriterijum. To možemo postići na više načina. Jedan od njih bi bio da, slično kao u unutrašnjem ciklusu, posebna vrednost ulazne promenljive označi da treba izaći iz ciklusa. Tako bismo mogli da uvedemo u naš program sledeće izmene: u liniji 150 umesto poruke „EVO NOVOG PITANJA" stavimo „ŽELIŠ LI JOŠ PITANJA?" i uvedemo nove linije:

```
152 INPUT Y$
154 IFY$=„DA" GOTO 160
156 STOP
```

Ako želimo da se postavi tačno određen broj pitanja, recimo 10, možemo ostvariti

Izlaz u slučaju opasnosti

Opasnost se zove: loš program

Ponekad je potrebno prekinuti odvijanje programa i u listingu ispraviti neku grešku. Na ovoj maloj prepreci mnogi se početnici sapliću, naročito ako je računar u INPUT modu, to jest ako očekuje niz ulaznih podataka. Evo kako treba postupiti u ovakvoj situaciji:

Spektrum

Prvo pritisnite CAPS SHIEFT; SPACE za prekid programa. Ako ovo ne radi, nalazite se u input modu. Koristeći kursor kontrolu i DELETE, odstranite jedan znak navoda. Zatim upotrebite STOP (šifrovano A), pa ENTER. Ovo proizvodi poruku „prestani sa ulazom". Zatim ponovo pritisnite ENTER za listanje programa.

Komodor

Pokušajte sa RUN/STOP posle čega treba otkucati LIST. Ako ne pomaže, pritisnite RUN/STOP i RESTOR u isto vreme, a zatim otkucajte list.

Acorn

Pritisnite ESCAPE, pa otkucajte list. Ako ne pomaže, pritisnite BREAK tipku, a zatim otkucajte OLD pa LIST.

Galaksija

Prvo pritisnite BRK pa otkucajte LIST. Ako ovo ne pomaže, pokušajte sa STOP/LIST.

brojački ciklus i bez posebnih instrukcija FOR . . . NEXT. Promenljiva B koja će predstavljati brojač ciklusa treba na početku da dobije vrednost 0, a svakim prolaskom kroz ciklus da se uvećava za jedan. Iz ciklusa se izlazi kada dostigne svoju graničnu vrednost, u našem slučaju 10. Za ostvarenje ovakvog brojačkog ciklusa treba da proširimo polazni program sledećim naredbama:

```
55 B=0
85B=B+1
150 PRINT „TAČNO,";X%
155 IF B=10 THEN STOP
```

Napomena: ako radite sa „galaksijom" u liniji 155 izostavite službenu reč THEN.

Brojevi pod kontrolom

Pošto unesete program i proverite da li radi, snimite ga i isključite računar. Prelazimo na eksperiment kojim ćemo pokušati da ustanovimo zakonitost po kojoj se generišu slučajni brojevi. Uključite računar, učitajte program, startujte ga i zapišite prvih nekoliko zadataka. Zatim isključite računar i ponovite celokupni postupak. Primetićete da je računar oba puta postavio isti niz pitanja. Činjenica je da se inicijalizacija slučajnih brojeva vrši uvek na isti način, jer se uvek polazi od istog početnog slučajnog broja. Stoga treba „uslučajniti" izbor prvog slučajnog broja, što se najčešće čini korišćenjem sistemskog časovnika. Tako mnogi računari, među kojima i „spektrum", koriste posebnu naredbu RANDOMIZE čija je namena da u bajtove odakle se uzimaju slučajni brojevi postavi neku „nepredvidivu" početnu vrednost, tako da nadalje, bez obzira što se koriste ista pravila za generisanje drugih brojeva iz niza pseudoslučajnih brojeva, ipak imamo slučajnu sekvencu brojeva.

Sličan efekat postiže se kod „komodora" i Acorna korišćenjem RND(0). Sasvim suprotno dejstvo kod ova dva računara ima navođenje negativnog argumenta RND funkcije. Za negativnu vrednost argumenta dobija se uvek ista vrednost funkcije. Možda se pitate čemu to može da služi. Upravo ova mogućnost može se korisno upotrebiti prilikom otkrivanja logičkih grešaka u programu.

Od prvog metoda poznatog pod imenom „sredine kvadrata" do danas predloženo je mnoštvo raznih tehnika za generisanje slučajnih brojeva. Da proverite kako to radi vaš računar, sastavite ovakav program. Neka računar generiše 1000 slučajnih brojeva od 1 do 5. U programu prebrojte učestalost pojavljivanja svakog od njih. Ako je korišćen dobar metod, sve ove frekvence treba da budu oko 200 jer je 200/1000 teoretska verovatnoća pojavljivanja svakog od ovih brojeva. Ako postoje velika odstupanja, onda primenjen metod nije dobar . . .

Nevenka Spalević

katalog kompjutera '85

kompletan hardver • monitori
disk-jedinice • računari •
ostali periferali • štampači

Celokupna svetska produkcija mikro-računara i kompletnog hardvera na jednom mestu! Gde? Na nekom sajmu elektronike? Ne, nego u prvom Jugoslovenskom KATALOGU KOMPJUTERA '85.

Odlučili ste da kupite svoj prvi kompjuter? Koji? Nema dileme: pomoći će vam prvi YU KATALOG KOMPJUTERA '85.

Ubrzo posle toga poželeli ste da na svoj računar priključite disk-jedinicu ili štampač? Ništa lakše: i drugi put pomoći će vam prvi YU KATALOG KOMPJUTERA '85.

Vi ste već iskusan haker i neophodan vam je savršeniji računar? Ma, sve je u redu: i treći, i svaki naredni put vaš najbolji savetnik biće prvi YU KATALOG KOMPJUTERA '85.

Iz sadržaja kataloga:
– kompjuterski rečnik
– fotografije svih modela
– tehnički podaci
– opis i opšte karakteristike
– cene proizvoda kod nas i u svetu
– saveti i preporuke za kupovinu
– adresa proizvođača i zastupnika u SFRJ

PRVI JUGOSLOVENSKI KATALOG LIČNIH I KUĆNIH KOMPJUTERA I KOMPLETNOG HARDVERA SVIH SVETSKIH PROIZVOĐAČA

Najnovije

CENA 600. DIN.
ISPORUKA POUZEĆEM.

Naručite na adresu: KATALOG KOMPJUTERA '85. 34000 KRAGUJEVAC.

VREME RADI ZA ONE KOJI PRATE SVE NOVO

U knjižarama i papirnicama Mladinske knjige uvjek ima šta novo: nove knjige i sve širi izbor programske i druge računarske opreme:

6 novih priručnika za vlasnike Spectruma i Commodora:

1. THE COMPLETE SPECTRUM (488 stranica)	3900
2. AN EXPERT GUIDE TO THE SPECTRUM	1800
3. THE SPECTRUM GAMESMASTER	1600
4. THE COMPLETE COMMODORE 64 (488 stranica)	3900
5. ADVANCED MACHINE CODE PROGRAMMING FOR THE C64	2200
6. USEFUL SUBROUTINES AND UTILITIES FOR THE C 64	1800

Nova izdanja na srpskohrvatskom jeziku:

19. Grupa autora: KUĆNI KOMPJUTERI algoritmi i programi	780
20. Marković, Davidovac: ZX SPECTRUM programiranje u BAZIC-u	750
21. Pasarić: ZX SPEĆTRUM — uvod u rad i programiranje	300
22. Grupa autora: SPEKTRUM PRIRUČNIK	1200
23. „Polo": ZX SPECTRUM — upotreba i programiranje	1250
24. Čip, Šahinpašić: KOMPJUTERSKA POČETNICA	680
25. Šavić, Gačić: PRIMENA MINI RAČUNARA	490
26. Špiler: BASIC (prevod — novo izdanje) .	1150,

Neka zanimljivija nova izdanja na slovenačkom jeziku:

27. COMMODORE 64 — priročnik za uporabo (prevod)	980
28. KASETNA ENOTA VC 1530/VC 1531 (prevod)	220
29. Špiler: BASIC ZA ZX SPECTRUM	1500
30. Jakopin: INES — urejevanje podataka, tekstova i slika — priručnik s kasetom	1500
31. PROGRAMI ZA ZX SPECTRUM — posebno izdanje revije „Moj mikro"	1100

Još uvek najtraženije kasete s programima za Spectrume:

32. MAČAK MURI BROJI I RAČUNA	900
33. KONTRABANT 2 (prevod)	1300
34. KASETA RADIA ŠTUDENT (Kontrabant 1+9 programa — prevod)	1300

Uskoro će biti rasprodati — 8 engleskih priručnika iz prve serije:

7. THE ZX SPECTRUM AND HOW TO GET THE MOST FROM IT	1500
8. SPECTRUM GRAPHICS AND SOUND	1750
9. THE SPECTRUM BOOK OF GAMES	1500
10. DATA HANDLING ON THE COMMODORE 64 MADE EASY	1500
11. COMMODORE 64 GRAPHICS AND SOUND	1750
12. BUSINESS SYSTEMS ON THE COMODORE 64	1750
13. COMODORE 64 DISK SYSTEMS AND PRINTERS	1500
14. 6502 MACHINE CODE FOR HUMANS	2000

I još dva nova priručnika i rječnika za ozbiljniju upotrebu:

15. A PARENT'S GUIDE TO EDUCATIONAL SOFTWARE FOR COMPUTERS AT HOME AND IN THE SCHOOL.	1200
16. HOW TO CHOOSE AND USE BUSINESS MICROCOMPUTERS AND SOFTWARE	1200
17. THE CONCISE ENGLISH DICTIONARY	5000
18. SIMPLE ENGLISH DICTIONARY	980

U našoj knjižari s prvim specijaliziranim računarskim odeljenjem u Jugoslaviji možete kupiti i **profesionalne tastature za spectrume** (INES i još dva modela), **Joysticke za Spectrume i Commodore, Printerface** (interface za povezivanje spectruma s štampačem), **dva štampača** (RX-80 i RX-100) i **druge proizvode firme EPSON, kablove za povezivanje računara i kasetofona** (PIN—DIN), **stalke za Spectrume i diskete, prazne kasete za snimanje programa** itd.

AKO SE BAVITE RAČUNARIMA NE MOŽETE MIMOIĆI MLADINSKU KNJIGU!

Ispunjenu narudžbenicu — za pouzeće ili overenu od RO — pošaljite na našu adresu:

KNJIGARNA MLADINSKE KNJIGE,
61000 Ljubljana, Titova 3 (061 211-895)

NARUDŽBENICA R—5

Potpisani (ime i prezime — adresa RO) _____

Tačna adresa _____

Neopozivo naručujem (pouzećem — za potrebe RO) sledeće knjige: _____

_____

_____

Datum: Potpis (žig RO):

mladinska knjiga
knjigarne in papirnice

Hakeri u nevolji

SOS

Zahvaljujući dugo vremena zabranjenom uvozu i ostalim restriktivnim merama, mnogi vlasnici kućnih kompjutera u Jugoslaviji zapali su u vrlo neugodnu situaciju: prvo nekako nabave računar, a onda ostanu bez knjiga, uputstva ili stručne pomoći. U želji da pomognemo svima koji su zapali u teškoće, rešili smo da pokrenemo rubriku u kojoj ćemo davati odgovore na sva vaša pitanja. Pišite na adresu: „GALAKSIJA" — rubrika „SOS" — Bulevar vojvode Mišića 17, 11000 Beograd. U očekivanju vaših pisama, dajemo odgovore na pitanja koja su već ranije stigla u redakciju.

Simulacija CALL (HL)

Zoran Milovanović iz Zagreba pita kako da ostvari mašinsku naredbu CALL (HL), koja, naravno, ne postoji u setu instrukcija procesora Z80.

Naredba CALL (HL) se može ostvariti na sledeći način:

```
LD    DE, NASTAVAK
PUSH DE
JP    (HL)
NASTAVAK . . .
```

Naredba CALL mora, najpre, da stavi na stek povratnu adresu. To je postignuto sa:

```
LD    DE, NASTAVAK
PUSH DE
```

Zatim naredba JP (HL) nalaže procesoru da skoči na adresu sadržanu u HL registru. Kad se potprogram završi, naredba RET će skinuti sa steka adresu labele „NASTAVAK" i naložiti procesoru da nastavi sa radom odatle.

Nevolja ovog postupka je u tome što zahteva korićenje DE registra.

Naredba CALL (HL) se zato, može ostvariti i na sledeći način:

```
LD    (XX+1), HL
XX CALL 0
```

Naredba LD (XX+1), HL stavlja adresu iz HL registarskog para na mesto one nule iza CALL naredbe, zatim se izvršava CALL naredba koja skače na zadatu adresu. Umesto HL, mogu se koristiti registri BC ili DE, što bi odgovaralo naredbama CALL (BC), odnosno CALL (DE).

Priručnici za mašinac

Mnogi nas pitaju koja je najbolja knjiga za učenje mašinskog jezika za procesore Z80 i 6502.

Verovatno najbolje knjige koje se mogu nabaviti su „PROGRAMING THE Z80 A", odnosno „PROGRAMING THE 6502". Autor: Rodnay Zaks, izdavač: SYBEX Ltd, Unit 4, Borrne Industrijal Estate, Crayford, Kent Dal 4BU, England. Cene su, na žalost, dosta visoke, oko 15-tak funti.

Programi koji sami sebe pišu

Zoran Milojković iz Beograda pita na koji način rade programi koji sami sebe pišu i da li postoje na bejziku (spectrum).

Pre svega, program ne može sam sebe da napiše(to bi bilo isuviše!) ali može sam sebi da doda neku programsku datoteku koju je upravo izračunao, ili promeni neku programsku liniju, zavisno od toga kakav se posao u datom trenutku obavlja. Neke mašine su dušu dale za takve stvari, recimo „galaksija", ali „spektrum" se vrlo teško koristi u tu svrhu, pre svega zbog gomile sistem promenljivih koje kontrolišu bejzik i zbog komplikovane raspodele memorije. Ipak, na raspolaganju je program beta bejzik (verzija 1. 8) koji, zahvaljujući naredbi KEYIN, omogućava takve manipulacije.

Point i Screen$

Mnoga pitanja vezana su za još uvek zagonetne naredbe POINT I SCREEN$ „spektrumovog" bejzika.

Funkcija POINT (X, Y) kao rezultat daje 1 ili 0, zavisno od toga da li je tačkica na ekranu, čije su koordinate X i Y, osvetljena ili nije. Funkcija SCREEN$ (Y, X) kao rezultat daje slovo koje se nalazi na ekranu, na koordinatama Y, X. Na primer: PRINT SCREEN$ (10, 10) će odštampati „A", ako se na poziciji 10, 10 nalazi slovo „A". Ove dve naredbe naročito dolaze do izražaja kod programa koji ispituje sliku na ekranu. Da ih nema, morao bi da se dimenzioniše niz koji bi pamtio sadržaj ekrana. SCREEN$ se, naravno, koristi i za snimanje ekrana na traku. (SAVE „ime" SCREEN$).

Mikrać za fortran i kobol

Marko Bernić iz Zadra pita za najjeftiniji kućni kompjuter koji bi mogao da radi na fortranu ili kobolu?

Po svoj prilici, to je „epl II", naravno sa perifernom opremom. Ipak, imajte na umu da kobol dolazi do izražaja tek kad se radi sa gomilom datoteka i diskova, a da skoro sve što može fortran, može i neki bolji bejzik.

Koji je računar bolji

Čedomir Stojanović iz Obrenovca želi da nabavi računar i pita koji je od tri modela koji ga interesuju najbolji — „atari 800 XL", „komodor 64" ili „amstrad"?

„Atari 800 XL" je, po našoj oceni, nešto bolji od „komodora". Međutim, za „komodor" se kod nas može nabaviti veliki broj vrlo jeftinih programa, što nije slučaj sa „atarijem". Ako kupujete „komodor" ili „atari", pazite da ne zaboravite specijalni kasetofon.

Što se tiče „amstrada", on je verovatno najbolji računar koji se trenutno može kupiti u ekonomskoj klasi. Nevolja sa „amstradom" je jedino u tome što mu se cena ne uklapa baš najbolje u trenutne carinske propise.

Slobodni i zabranjeni registri

Miloš Petrović iz Beograda se interesuje da li sme da koristi IX, IY i alternativni set registra u mašinskom programu koji se poziva sa RANDOMIZE USR naredbom (spectrum).

Indeksni registri mogu da se koriste bez problema, s tim da u IY mora da se nalazi broj 5C3A (heksadekadno), ako mašinski program poziva rutine iz ROM-a. Alterativni set registra takođe može da se koristi bez problema, s tim da pri povratku u basic HL' mora da sadrži broj 2758 (heksadekadno). U protivnom, može se desiti da bejzik program u memoriji bude „nafilovan" nekim čudnim stvarima.

Američki distributer

Jovan Atanaković iz Beograda traži da objavimo adresu nekog američkog distributera elektronskih delova.

Odlučili smo se za „Jameco". Adresa je: Jameco Electronics, 1356 Shoreway Road, Belmont, CA 94002, U.S.A. Za adrese koje objavljuje redakcija ne može da garantuje kvalitet usluge.

Očitavanje tastature

Dušan Jokić iz Smedereva, koji je tek zašao u mašinsko programiranje (spectrum), želi da mu pomognemo kako da ispita da li su dva tastera (bilo koja, a ne u grupama od po pet) pritisnuta u isto vreme.

To se najlakše može rešiti pozivanjem rutine iz ROM-a na adresi 028E (heksadekadno). Ako je pritisnut samo jedan taster, onda će njegov kod biti smešten u E registar. Ako su pritisnuta dva tastera, onda će kod jednog biti smešten u D a kod drugog u E registar. Najzad, ako nije pritisnut nijedan taster, onda će u D i E registre biti smešteni brojevi 255. Imajte na umu da u ovom slučaju kodovi nisu po ASCII standardu.

Ukucajte sledeća dva kratka programa (jedan na bejziku, drugi na mašincu) da se uverite kako to radi.

```
5 CLEAR 40000
10 LET X=USR 50000:
20 PRINT AT 10,5; „D="; INT(X:256); TAB 12;
„E=";X-256 *(X:256);„ „„ "
30 GO TO 10
```

```
        ORG     50000
CD8E02  CALL    #028E
D5      PUSH    DE
C1      POP     BC
C9      RET
```

Mašinac na TI 99/4A

Ivo Grospić iz Splita, vlasnik kompjutera TI 99/4A, pita na koji način može da radi na mašinskom jeziku i koji procesor poseduje TI — 99/4A.

TI — 99/4A, na žalost poseduje takozvani zatvoreni operativni sistem, dakle bez naredbi PEEK, POKE i USR, tako da je mašinsko programiranje praktično nemoguće. Jedino prihvatljivo rešenje predstavlja nabavka modula „ASSEMBLER". TI — 99/4A poseduje šesnaestobitni procesor TMS 9900.

Priprema
Vladimir Kostić

Računari u domaćoj radinosti

priča o finoj grafici

Računar „Galaksija"

Ideja za grafiku visoke rezolucije nastala je još prošlog leta kao rezultat razmišljanja o tome da li kôdovi znakova mogu da se pošalju direktno na ekran u obliku niza bitova, umesto što se u EPROM-u 2716 transformišu u lik tog znaka? Razmatrana sa više detalja, ideja je odvela do jednostavnog sklopa koji bi trebao da je realizuje. Autori su otišli i korak dalje: razvijena su dva različita sklopa, koja su vršila istu funkciju. Prvi ima četiri integrisana kola, dok drugi, usavršeni, ima samo dva. No, o tome će više reći biti u narednom broju. Biće objavljena kompletna shema sa detaljnim uputstvom za samogradnju i neophodne prepravke na „galaksiji".

Problem po problem . . .

Realizacija „hardvera" bila je samo jedan deo posla, i to onaj manji. Trebalo je još sastaviti program koji će vršiti sâm prenos bitova iz memorije na ekran. Da bi se dobila stabilna slika, taj program mora da zadovolji vrlo stroge kriterijume u pogledu brzine izvršavanja, pa čak i u pogledu izbora mašinskih instrukcija kojima će biti realizovan. Među „zabranjenim" instrukcijama našle su se i sve one kojima Z-80 komunicira sa memorijom. To je, praktično, značilo da se tokom crtanja slike svi parametri moraju nalaziti u internim registrima samog mikroprocesora.

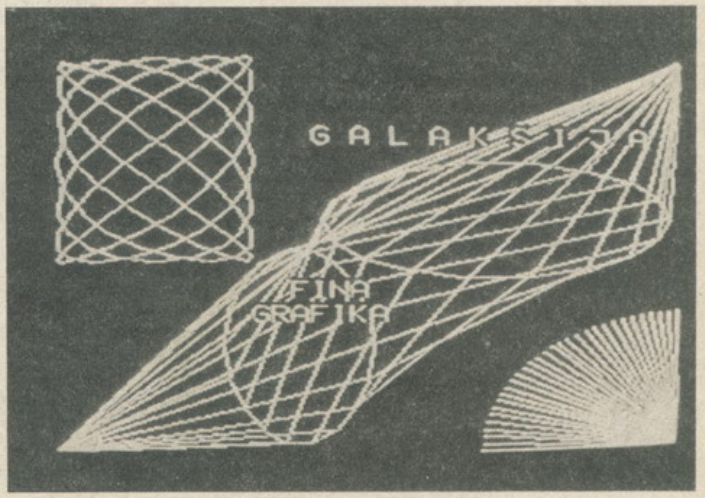
Čim je taj problem rešen, pojavio se sledeći: imamo grafiku, ali šta s njom da radimo? Vlasnicima „komodora 64" poznato je da ne mogu da rade sa tekstom kad uključe grafiku i obrnuto, što je inače vrlo neprijatno. Rešenje koje se nametnulo samo po sebi je da se tekst, slikovito rečeno, iscrtava u grafičku memoriju. Pošto se generator znakova „galaksije" ne nalazi u adresnom prostoru mikroprocesora, bilo je neophodno formirati novi set slova. U tome nam je važnu ulogu odigrao Ljuba Gavrilović, koji je taj dosta neprijatni posao preuzeo na sebe. Ovoga puta, u RAM-u je ostavljen pokazivač početka generatora znakova, čime je omogućeno menjanje setova slova. Sada je moguće u ROM ili RAM staviti ćirilicu, grčki alfabet ili neke proizvoljne oblike formirane u matrici $(7+1)\times 9$. Pošto se slova iscrtavaju u bit mapu, svi setovi slova mogu se u isto vreme nalaziti na ekranu. Na žalost, povratkom u režim za prikaz teksta svi znakovi koji su bili na ekranu postaće latinični, tačnije ASCII, već prema upotrebljenom kôdu.

Kao što obično biva, što se ide dalje, problema je sve više. Sada je trebalo prepoznati svaki pokušaj „galaksije" da napiše neko slovo na ekranu i to slovo nacrtati na odgovarajućem mestu u bit mapi. Link za video je bio rešenje u svim slučajevima, osim u jednom: editovanje neke programske linije ne ide preko linka za video i nije se moglo detektovati ni na koji način. Bila je to kap koja je prelila već davno punu čašu — i „galaksija" je dobila ekranski editor.

Ekranski editor

Uzor za izradu ekranskog editora bio je „komodor 64". Kurzor koji trepće (da bi bio upadljiviji i da bi se videlo slovo ispod njega) može da se pomera po celom ekranu upotrebom strelica i tastera REPT. Mogućnosti koje se time otvaraju su velike i uglavnom poznate. Autorima je najveće olakšanje predstavljalo to što linije sa greškom ne moraju da se kucaju ponovo: dovoljno je dovesti kurzor do mesta greške, napraviti ispravku i pritisnuti ENTER. Višak znakova briše se tasterom DEL (strelica ulevo će samo pomeriti kurzor), dok se manjak nadoknađuje pritiskom na SHIFT i O. Jednostavno, to je jedina kombinacija tastera kojima je mogla biti dodeljena funkcija INSERT. Pri tome će na mesto kurzora biti ubačeno blanko, a ostatak linije će biti pomeren udesno.

Ništa na svetu nije savršeno, pa ni „galaksijin" ekranski editor: kada dužina nekog reda na ekranu dostigne 32 znaka (kada se ceo red popuni), on se automatski povezuje sa sledećim redom u logičku celinu. To je bilo neophodno da bi se omogućilo pisanje programskih linija dužih od 32 znaka, ali zato kod ubacivanja znakova pomoću SHIFT O (INSERT) može doći do neželjenog povezivanja sa sledećim redom koji predstavlja odvojenu celinu.



Ovaj artefakt može da ima i korisnu primenu, jer omogućava jednostavno nadovezivanje jedne ili više programskih linija, koje čak i ne moraju u programu da slede jedna za drugom.

Editoru je dodata još jedna korisna stvar (inspiracija CBM-64): kada se otvare znaci navoda, svaki pritisak na neku od strelica proizvodi kontrolni karakter. Kod kasnijeg štampanja alfanumerika, kontrolni kôdovi imaju efekat kao da je u neposrednom režimu (van navodnika) pritisnuta strelica. Ovaj dodatak otvara mogućnosti formatizovanja ispisa na ekran.

Grafičke naredbe

Zajedno sa ekranskim editorom, „galaksija" je dobila i nove grafičke naredbe u bejziku. Prva od njih je naredba GRAPH (bez parametra) koja uključuje grafički mod. Ta naredba je dodata da bi se omogućila ušteda memorije: grafika „troši" 6,5 K (tačnije 6688 bajtova) koji mogu zatrebati u većim programima. Tu je, naravno, i naredba TEXT (takođe bez parametra), koja isključuje grafički mod. Kada je grafika isključena, „galaksija" generiše sliku na svoj uobičajeni način, dok ekranski editor ostaje i dalje na raspolaganju.

Za samo crtanje obezbeđene su četiri naredbe. PLOT X, Y i UNPLOT X, Y osvetljavaju, odnosno zatamnjuju tačku sa koordinatama (x,y), koja ujedno postaje i nova pozicija fiktivnog grafičkog „pera". Pošto je rezolucija slike 256×208, vrednosti koordinata moraju da budu u opsegu od 0 do 255 za x, odnosno od 0 do 207 za y koordinatu. Ove vrednosti se uzimaju po modulu 256, tako da prekoračenje opsega praktično nije ni moguće. Vrednosti y koordinate veće od 207 a manje od 256 smatraju se nulom. Ekran je smešten u prvi kvadrant koordinatnog sistema, što znači da se tačka sa koordinatama (0,0) nalazi u donjem levom uglu ekrana.

Druge dve naredbe služe za povlačenje bele, odnosno crne linije od trenutne pozicije pera do naznačene tačke. To su naredbe DRAW X,Y i UNDRAW X,Y. Za interpoliranje prave linije po prvi put upotrebljen je algoritam Nenada Dunjića, jednog od autora. Taj algoritam pretenduje da bude (i najverovatnije jeste) najjednostavniji i, što je još važnije, najbrži način interpolacije prave linije poznat do sada. Zahvaljujući upotrebi Nenadovom brzom algoritmu, „galaksija" crta linije sasvim zadovoljavajućom brzinom i potpunom preciznošću.

„Crtaći" na „galaksiji"

Zbog različitih memorijskih konfiguracija, bit-mapa „galaksijine" grafike morala je da poseduje mogućnost „pomeranja". Zato

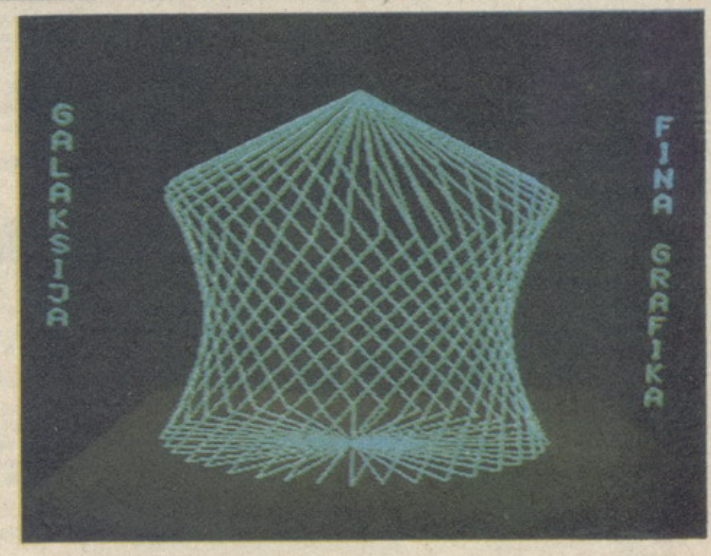
Kada je pre nešto više od godinu dana „galaksija" ugledala svet, bio je to prvi korak kojim su mnogi domaći ljubitelji računara ušli u novu eru. „Galaksiju" su svi prihvatili kao mezimče kome se, kao i svakom prvencu, ne gleda na mane. Kako je vreme, međutim, prolazilo, tako su i apetiti polako rasli. Memorija od 2, 4 pa i 6 K postala je tesna za programe, osnovni ROM previše skroman za umeća novopečenih programera, a grafički afiniteti su prevazišli skromnih 3000 tačaka. Nakon interfejsa 1, memorijskog proširenja, programatora eproma, ROM-a 2 i generatora zvuka „Galaksija, najzad, dobija i nešto u šta je malo ko verovao — grafiku visoke rezolucije (53248 tačaka) sa ekranskim editorom! Sa ovim dodacima mala „galaksija" i nije više tako mala. Ako se ova rešenja objedine u „galaksiju 2", naš računar će ponovo postati kraljica na domaćem tržištu.

Fina grafika na finom računaru: Tvorci novog proširenja za domaći prvenac: Nenad Dunjić i Milan Tadić

je u RAM-u ostavljeno 5 bajtova (ne pitajte zašto baš pet) čija se vrednost proračunava i postavlja prilikom izvršavanja naredbe GRAPH. Njihov sadržaj govori mikroprocesoru odakle mu počinje bit mapa u memoriji. Dovoljno je na odgovarajući način promeniti sadržaj tih bjatova i slika će se generasati sa drugog mesta u memoriji. Sama promena je zbog toga trenutna, t.j. između dva generisanja slike, tako da teorijski možete prikazati 50 različitih slika u sekundi. Zbog ograničenog memorijskog prostora, praktično je moguće imati samo 7 slika (već 7 slika potrošiće zamašnih 45,5 K). Čak i tako skroman broj omogućava izradu interesantnih „pokretnih" slika i figura. Ako vaše ambicije nisu usmerene u tom pravcu, možete imati jednu sliku rezolucije 256×1500 po kojoj ćete „šetati prozor" rezolucije 256×208 tačaka.

Za rad grafike i ekranskog editora bilo je neophodno dodati tridesetak bajtova sistemskih podataka. Da bi se održala potpuna kompatibilnost sa postojećim programima, jedino raspoloživo mesto bilo je iznad RAMTOP-a. Tamo su sakrivena čitava 32 bajata (ili 6688 kada radi grafika). U narednim brojevima biće dati detaljni opisi i programi koji će preporoditi i vašu „galaksiju".

Milan TADIĆ



mali oglasi

SPEKTRUM

• Super nove programe za „spektrum" prodajem: brax bluf, circus, ghostbusters, giants revenge, zombie, alien 8, battle cars, kokotoni wile, D-DAY, fall of rome, fort apocalipse, solo flight... Štrumpf software, Veljko Vuković, Sukoišanska 37, 58000 Split, 058/582-484

• Diamondsoft researc ltd. Veliki izbor od preko 700 fabričkih programa za vaš „spektrum". Narudžbe moguće i pojedinačno. Cena programa 50—70 din. + poštarina + cena kasete kao i u kompletu od 20 programa po vašem izboru za 1000 din. sa poštarinom. Nudimo vam: skripta bejzik i mašinac, neophodne pri radu na vašem računaru. Uverite se u našu profesionalnost i odlične usluge. Sa „Diamondsoftom" u korak sa svetskim top listama. Dodatne informacije na telefon: 021/20-742, katalog besplatan! Narudžbe šaljite na adresu: Veselinov Đorđe, Maksima Gorkog 6, 21000 Novi Sad

SPECTRUM — LITERATURA:
ZA POČETNIKE I SVE OSTALE — sve što vam treba za programiranje u bejziku (grafika, muzika i sve drugo) možete naći u jedinom kompletnom profesionalnom prevodu „spektrumovog" BASIC PROGRAMIRANJA " i brošure „UVOD". Kvalitetna ofset štampa, isporuka odmah pouzećem, za samo 880 din. Duško bjelotomić, Centar 1, 54550 Vaipovo, tel, 054/82-665 ili 041/683-141.

• GALACTIC SOFTWARE! Prodajem preko 500 programa za „spektrum." Cene veoma niske. Kvalitetni snimci. Moguća je razmena. Prodajem i profesionalne prevode literature za „spectrum" Javite se i tražite besplatan katalog.
IVAN NOŽINIĆ
21000 NOVI SAD
MAKSIMA GORKOG 6 021/26-518

• PRODAJEM najbolje i najjeftinije prevode: 1. „Sepktrum" bejzik programiranje (110 strana) 1000,00 dinara. 2. „Spektrum" mašinski jezik (170 strana) 1000,00 dinara. Zoran Čukalo, Potočnjakova 7/I, 41020 Zagreb

• Spectrum 48K veliki izbor klasičnih, novih i najekskluzivnijih programa, povoljno, na vašim ili našim kasetama. Sve informacije, spisak literature i besplatni katalog tražite preko celog dana telefonom ili na adresu: Bajić Goran, Stevana Filipovića 29/85, 11040 Beograd, tel. 011/653-285

• Spectrumovci, za sve one programe za koje ste čuli da postoje a ne znate kako da dođete do njih najpovoljnija ponuda na tržištu Yu-softwarea. besplatni katalog i ostale informacije. Javite se. Gusić Josip, Bulevar AVNOJ-a 117/3, 11070 Novi Beograd, tel. 011/146-173

Mislite li da ste dovoljno informisani o stanju na mikroračunarskoj sceni ako pročitate nekoliko časopisa mjesečno? Najaktuelnije iz evropskih kompjuterskih centara prenosimo u našim najnovijim prospektima. Opširnije u našem oglasu u rubrici „RAZNO".
EUROCATALOG, Vlajić Saša, Hasana Kikića 9, 7800 Banja Luka

• Stručnjak za programiranje daje časove mašinca. tel.: 400-823

• SPECTRUMOVCI — specijalno postolje za vaš kompjuter od kvalitetne plastične mase koje omogućava strujanje vazduha i odvođenje suvišne toplote sa vašeg SPECTRUMA, pri čemu tastatura zauzima funkcionalnan nagib za samo 1000 dinara. Dragan Dejković, Beograd, Ljube Didića 40/4, tel. 011/768-505.

• Spektrumovci — ekskluzivno i povoljno: veliki izbor klasičnih, novih, najnovijih programa, literatura, besplatni katalog. Bajić Goran, Stevana Filipovića 29/85, 1104 Beograd, Tel. 011/653-285

INFROMIRAJTE SE PRIJE NEGO ZAKORAČITE U SVET RAČUNARA
— Dopisna škola BASIC-a
— Prevodi uputstava i najboljih knjiga za „Commodore 64"
— Usluga obrada podataka za magisterij i doktorat
— Programi za „commodore 64" „RASUMI" TO ČINI KOMPLEKSNO.
Javite se na adresu:
„RASUMI", Computerlab
postfah 313, 54000 Osijek

COMMODORE

• Commodore 64 software pruža vam se prilika da nabavite po minimalnoj ceni 50 najboljih programa za vaš kompjuter! Pored toga, nudim vam izbor od preko 1000 ostalih programa! Sve što drugi nemaju potražite u besplatnom katalogu programa! Mirko Žagar, Vukasovićeva 82, 11090 Beograd, tel. 011/592-024 (od 15 do 18h)

• Pedeset najboljih programa za „komodor 64". Izvanredna prilika da po minimalnim cenama obogatite svoju kolekciju programa!
Mirko Žagar, Vukasovićeva 82, 11090 Beograd, tel. 011/592-024 (od 15—18h)

• Pored adaptera za priključenje svakog kasetofona na „komodoru 64", ovde možete nabaviti i mnoštvo programa. Cene — prava sitnica.
Ilić Vladimir, B. Kidriča 5/II, 22300 St. Pazova, tel. 022/311-013

• Imam samo najskuplje i najbolje programe. Javite se radi kupovine ili razmene. Branislav Bojadžievski. J. Gagarina 205 11070 N. Beograd tel. 011/156-444

• „Komodor 64", programi, noviteti. Uverite se u naš kvalitet. Povoljni uslovi. Niske cene. Popusti. Bogata selekcija. Besplatan katalog. Brzo odgovaram.
OLIVER VUJOVIĆ, Georgi Georgiju Deža 26/3, 11070 N. Beograd

RAZNO

• Za C64 nudim video rikorder VCR Philips N-1502, frekvencmetar 250 MHZ, generator funkcija, led obrtomjer. Šaljem spisak. Ostojić Nedeljko, Vj. B. Kidriča 27, 54000 Osijek

• Prodajem printer PC 100 C sa uputstvom za TI59/58 (3500000), 6 rolni papira (300000), maths/utilities modul sa uputstvom (750000). Sve: 4000000. Nikoloski Tome, Gradski zid, kula 23 stan 26, 91000 Skopje.

• Prodajem računar Sinclair QL. 011/558-007

ZX-81 & SPECTRUM
apple computer
commodore

KOPJUTERAŠI!!!
INFORMIŠITE SE NA VRIJEME!
Iz najelitnijih i najkompetentnijih njemačkih i engleskih magazina, te kataloga vodećih proizvođača računarske opreme izabrali smo ono što će vas najviše interesovati. Računari, periferije, najnovija literatura, software, sve cijene i adrese. Zbog obilja podataka nudimo vam dva prospekta:
1. SINCLAIR & COMMODORE (220 d)
2. APPLE (220 d)
Upoznajte se s mogućnošću kako da svoj Spectrum pretvorite u sintisajzer; najjeftiniji drajvovi za C-64; 7 novih kopija Apple-a II za svačiji džep, itd.
Isporuka pouzećem. Uvjerite se u naš kvalitet. Da budete u koraku sa Evropom brine se:
EUROCATALOG, Vlajić Saša, Hasana Kikića 9, 78000 Banja Luka.

Sve što je potrebno vašem ZX SPECTRUM-u možete dobiti od

COMET SOFTWARE

MILOVANOVIĆ LJUBIŠA
Petra Lekovića 57
11030 BEOGRAD
el: 0°1/558-007

LITERATURA
za sve MIKROPROCESORE
za sve MIKROKOMPJUTERE

COMET SOFTWARE

MILOVANOVIĆ LJUBIŠA
Petra Lekovića 57
11030 BEOGRAD
tel: 011/558-007

BYTE SHOP
package software

Trenutno najpopularniji način nabavke programa — paketi programa. Dve su bitne odlike ovakvog načina nabavke: izuzetno pristupačne cene i brz rok isporuke. Nazovite nas i već istog dana programi će brzinom svom, poštom krenuti u vaš dom!

1). 50 VELICANSTVENIH.... 3990 din. Kupili ste nedavno „spectrum"? Nije li najbolje da vašu kolekciju programa započnete upravo sa onim najboljima (čije opise možete naći na stranicama Računara

2) 10 ŠAHOVA +4 LOGIČKE IGRE... 900 din.

3) 50 BASIC PROGRAMA... 1500 din. Idealno za početnike! Igra + mogućnost neposrednog učenja basica. (svaku igru možete prekinuti, analizirati možda i poboljšati)

4), 5), i 6) 16K PROGRAMI.... 1500 din. svaki. Vlasnici „spektruma" 16K, Byte shop misli na vas, i nudi vam ova tri paketa od po 41 programa svaki. Jasno, moguće ih je koristiti i na „spectrumu 48K".

A od sada i veliki svetski hitovi i ostali programi u paketima po 14—16 programa za 1000 din. Mogućnost i pojedinačnog naručivanja. Gore navedene cene uključuju cenu kasete i poštarine. Tražite novi besplatni katalog paketa.
Adresa: OMRČEN DAMIR Bulevar JNA 64/10 11000 BEOGRAD telefon: 011/662-044

Kako **ostati** pilot — novi „pilotski bukvar" Zorana Modlija

„PISTE U NOĆI"

Nakon uspeha „Krilate katedre", profesionalni pilot i nastavnik letenja Zoran Modli nudi još jedan korak bliže tajnama letenja. Iako obrađuju za nijansu složenije probleme „slepog" letenja savremenih aviona u vazdušnom saobraćaju, „Piste u noći" takođe se čitaju bez daha!

IFR Snupi protiv VFR Barona: za i protiv instrumentalnog letenja, iluzije pilota, simulatori letenja.
Aladinove elektronske lampe: elektronski uređaji savremenih aviona.
Neko me posmatra: kontrola letenja na sve tešnjem nebu.
Ceo svet u torbi: ostavština kapetana Džepsena.
Muzika za AM lutalice: radio-navigacija.
Tamna strana neba: kako preživeti vremenske nepogode.
Poslednjih 20 kilometara: precizni i neprecizni instrumentalni prilazi.
Svedočanstva o udesima: ne vidim pistu, slećemo!
Kućni računar umesto aviona: obiđite svet ne izlazeći iz dnevne sobe.
Ja kapetan: odletite sami za Dubrovnik...

.... samo je deo izuzetno zanimljivog i (ne samo za pilota) upotrebljivog štiva, na nekoliko stotina strana i isto toliko ilustracija.

GALAKSIJA — BIGZ Bulevar vojvode Mišića 17 11000 Beograd

Ovim neopozivo poručujem _____ primeraka knjige „Piste u noći" u izdanju „Narodne knjige" iz Beograda, po pretplatnoj ceni od _____ dinara. Platiću prilikom prijema pošiljke — POUZEĆEM.

Ime i prezime _____

Adresa _____

Potpis _____ Datum _____

Naručite specijalna izdanja časopisa „Galaksija"

NARUDŽBENICA

GALAKSIJA, Bulevar vojvode Mišića 17, 11000 Beograd

Ovim neopozivo naručujem sledeća specijalna izdanja: 1. Računari broj 1, 2. Računari broj 2, 3. Računari broj 4, 5. Nikola tesla — neostvarena otkrića. Iznos od ukupno _____ dinara uplatiću prilikom preuzimanja pošiljke — pouzećem.

Ime i prezime _____

Ulica i broj _____

Broj pošte i mesto _____

_____ _____ _____

Datum Potpis

NAPOMENA: Ako ne želite da isecanjem oštetite svoj primerak „Galaksije", narudžbenicu prepišite na dopisnici ili u pismu.

ZABAVA★UČENJE★POTREBA

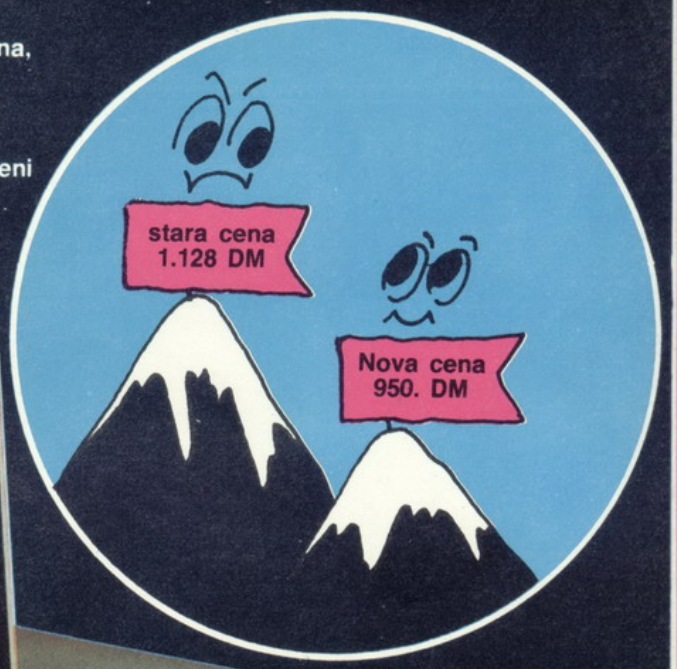
Iz serije 700, predstavlja vam s novom, nižom cenom, kompaktni personal kompjuter SHARP MZ — 731.

Kasetofon — Cetvorobojni štampač (crna, plava, crvena, zelena) — Kapacitet memorija 64 KB RAM.

Uz MZ 731 možete posebno naručiti:

„Zeleni" monitor po ceni od 420 DM ili TV kolor monitor po ceni od 840 DM.

Dinarske dažbine cca 65%.



SHARP

Mercator — Mednarodna trgovina

Za sva detaljna obaveštenja obratite se:

TOZD CONTAL
61 000 Ljubljana
Titova 66
Telefon: 061/328 441 lokal 939